UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

MODELAÇÃO E IMPLEMENTAÇÃO DE PROCESSOS NA PLATAFORMA *WEBMETHODS*

Rui Emanuel Brito de Almeida

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2011

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

MODELAÇÃO E IMPLEMENTAÇÃO DE PROCESSOS NA PLATAFORMA *WEBMETHODS*

Rui Emanuel Brito de Almeida

ESTÁGIO

Trabalho orientado pelo Prof. Doutor João Pedro Guerreiro Neto

e co-orientado pelo Engenheiro José Pedro Cardoso

MESTRADO EM ENGENHARIA INFORMÁTICA

Especialização em Sistemas de Informação

2011

Agradecimentos

Em primeiro lugar, onde estiveres mãe, consegui… É tudo graças a ti mãe, obrigado!

Quero agradecer a Indra por me ter recebido no âmbito do projecto de estágio e por me ter ajudado em todos os momentos menos bons.

Um agradecimento especial à Rute Arez, João Cordeiro e João Ramos pelo apoio, compreensão e preocupação que tiveram, mostraram-me que a Indra além de ser uma grande empresa também têm um grande lado humanitário.

Agradeço a todos os meus colegas de equipa por me terem acolhido de braços abertos em especial à Mónica Valverde, Steve Fernandes e ao João Felisberto pela excelente entreajuda e ambiente de trabalho.

Quero agradecer a toda a minha família e amigos, principalmente á minha namorada e aos pais da minha namorada pelo apoio prestado durante todo o estágio, sem eles não tinha conseguido. Um agradecimento especial ao meu tio José Brito por me ter dado bons conselhos na minha entrada na Faculdade de Ciências.

Agradeço ao Professor Dr. Luís Costa e Dr. Rui Esteves do Hospital de Santa Maria que fizeram o impossível para que a minha mãe visse o fim do meu curso... E conseguiram, obrigado!

Agradeço ao Bruno Branco e Brito, Bruno Correia, David Silva, João Casais, Pedro Martins, Rui Ferreira e à Sara Patrocínio, entre outros amigos pelas noitadas na faculdade (grandes noites) durante a Licenciatura e Mestrado onde brincávamos, riamos, discutíamos e por incrível que pareça também fazíamos os projectos.

E por fim quero agradecer ao Professor João Neto e ao meu coordenador Engenheiro Pedro Cardoso pela disponibilidade e empenho para que conseguisse atingir os meus objectivos durante o estágio.

Para vocês mãe e avó...

Resumo

O presente documento tem como objectivo demonstrar o trabalho que foi desenvolvido no âmbito do estágio pertencente à unidade curricular Projecto em Engenharia Informática.

O estágio pretende focar-se principalmente na integração, na modelação e na implementação de processos de negócio num contexto real de uma organização: o registo de dispositivos por parte de entidades.

Toda esta modelação e implementação de processos serão efectuadas sobre a plataforma *WebMethods*, uma ferramenta proprietária da empresa *Software AG*.

Em termos técnicos serão identificados alguns requisitos essenciais ao desenvolvimento deste estágio:

- ✓ Arquitectura Orientada a Serviços (SOA);
- ✓ *WebMethods Composite Application Framework* (CAF) – camada de apresentação disponibilizada em forma de *portlets*, executada sobre *My WebMethods Server*;
- ✓ *Web Services*.

Como objectivos principais deste estágio salientam-se:

- ✓ Efectuar o levantamento funcional de negócio de uma das áreas da organização;
- ✓ Modelar os processos de negócio na ferramenta de modelação da plataforma *WebMethods*;
- ✓ Efectuar o desenho técnico do modelo físico e dos serviços que cada passo do processo necessita para a sua correcta execução;
- ✓ Implementação dos processos e respectivos serviços;
- ✓ Executar os casos de testes definidos para validação funcional e para realização de testes de aceitação da aplicação.

Palavras-chave: *WebMethods*, *Business Process Management*, Aplicações *Web*, Sistemas de Informação e Bases de Dados e Arquitectura Orientada a Serviços

Abstract

This document aims to demonstrate the work that was developed during the second year of the Informatics Engineering MSc.

The aim of this training course is to focus mainly on integration, modeling and implementing business processes in a real context of an organization: the registration of devices.

All this modeling and implementation process will be carried out on the webMethods platform, a proprietary tool company Software AG.

In technical terms are identified some requirements essential to the development of this stage:

- ✓ Service-Oriented Architecture (SOA);
- ✓ WebMethods Composite Application Framework (CAF) - the presentation layer available in the form of portlets running on My webMethods Server;
- ✓ Web Services.

The main objectives of this stage show:

- ✓ Carry out a survey of business functional areas of an organization;
- ✓ Modeling business processes in the modeling tool WebMethods platform;
- ✓ Make a blueprint of the physical model and the services that each step of the process requires for its proper implementation;
- ✓ Implementation of processes and their services;
- ✓ Run the test cases defined for functional validation and acceptance testing of the application.

Keywords: WebMethods Business Process Management, Web Applications, Information Systems and Databases, and Service-Oriented Architecture

Conteúdo

Capítulo 1 Introdução 1

1.1 Contexto empresarial – *Indra* 1

1.2 Integração 2

1.3 Motivação e apresentação do problema 3

1.4 Objectivos 3

1.5 Organização do documento 4

Capítulo 2 Planeamento Técnico 7

2.1 Ferramenta *WebMethods* 7

2.1.1 *WebMethods Designer* 8

2.1.2 *WebMethods Developer* 9

2.2 Camadas lógicas 9

2.2.1 Camada de apresentação 11

2.2.2 *Portlets* 11

2.2.3 Camada de lógica de negócio 12

2.2.4 Camada de dados 13

2.2.5 Arquitectura tecnológica 13

2.2.6 Arquitectura física 15

2.3 Modelo de dados 16

2.3.1 Integraçao do modelo de dados 16

2.3.2 Modelo de dados RegD 17

2.4 Serviços do IS 17

2.5 Integrações a serem desenvolvidas com sistemas externos 18

2.5.1 *Reference data* 18

2.5.2 Repositório de entidades 18

2.5.3 Gestão documental 18

2.5.4 *Gateway* de pagamentos 18

Capítulo 3 Metodologias e planeamento 21

3.1 Método Indra para Desenvolvimento, Adaptação e Serviços - MIDAS .. 21
3.2 Planeamento do projecto 22
3.3 Recursos 23
3.4 Trabalho realizado 23
3.4.1 Implementação da *Milestone* 2- pedido de certidão 24
3.4.2 Implementação da *gateway* de pagamentos 24
3.4.3 Análise funcional de requisitos da *Milestone* 1 – registo de entidade 25
3.4.4 Implementação da *Milestone* 1 – registo de entidade 25
3.4.5 Validação e testes 25
3.4.6 Relatório final 26
Capítulo 4 Trabalho desenvolvido 27
4.1 Desenvolvimento em *WebMethods* 28
4.1.1 Desenvolvimento do processo BPM 28
4.1.2 Desenvolvimento e implementação de serviços no *IS* 28
4.1.3 Desenvolvimento e implementação das tarefas em CAF 29
4.2 Implementação do processo de pedido de certidão (*Assignment* 2) 31
4.2.1 Âmbito e especificação do processo 31
4.2.2 Desenho e descrição 31
4.2.3 Requisitos 35
4.2.4 Modelo de dados 38
4.2.1 Desenvolvimento do processo BPM 40
4.2.2 Desenvolvimento e implementação dos *Web-Services* 41
4.2.3 getCertificatesByProcessId 44
4.2.4 getDeviceInfoList 47
4.2.5 ValidatePayment 49
4.2.6 Desenvolvimento e implementação das tarefas 51
4.2.7 Tarefa de análise de director 51
4.2.8 FoCreditCardPayment 52

4.2.9 *Scheduler* de pagamento 54
4.3 Implementação do processo de registo de utilizador (*Milestone* 1) 56
4.3.1 Âmbito e especificação do processo 56
4.3.2 Desenho e descrição 56
4.3.3 Requisitos 57
4.3.4 Modelo de dados 60
4.3.5 Desenvolvimento e implementação dos *Web-Services* 63
4.3.6 checkIfEntityExistsInRegD 65
4.3.7 ManageUserCredencials 66
4.3.8 Desenvolvimento e implementação das tarefas 67
4.3.9 Integração com *Apache Directory* 69
4.4 *Gateway* de pagamentos 71
4.4.1 Arquitectura geral 71
4.5 Plataforma Comum 72
4.5.1 Módulo de transferência bancária 73
4.6 Módulo de multibanco 76
4.7 Cartão de crédito 78
Capítulo 5 Conclusão 83
5.1 Dificuldades encontradas e competências adquiridas 83
5.2 Trabalho futuro 83
Capítulo 6 Bibliografia 85

Lista de Figuras

Figura 1 – Organigrama da *Indra* 2

Figura 2 – Camadas Lógicas 10

Figura 3 – Estrutura do pacote no IS 12

Figura 4 – Arquitectura tecnológica 14

Figura 5 – Infraestrutura física 16

Figura 6 – Estrutura MIDAS 22

Figura 7- Input e Output do serviço 30

Figura 8- Conector com a estrutura do serviço 30

Figura 9 – Pedido de certidão 34

Figura 10 – Diagrama do pedido de certidão 38

Figura 11 – Modelo relacional 39

Figura 12 – Processo de pedido de certidão 40

Figura 13 – Estrutura dos documentos usados no pedido de certidão 42

Figura 14 – Documento de pedido de certidão 42

Figura 15 – Documento de pagamento 42

Figura 16 – Serviços principais 43

Figura 17 – Serviços de acesso à *gateway* 43

Figura 18 – Parâmetros de entrada 44

Figura 19 – Parâmetros de saída 44

Figura 20 – Estrutura do serviço 45

Figura 21 – Parâmetros de entrada 45

Figura 22 – Estrutura do serviço notificationEmail 46

Figura 23 – Pipeline sendEmail 47

Figura 24 – Parâmetros de entrada 47

Figura 25 – Parâmetros de saída 48

Figura 26 – Estrutura do serviço getDevicesInfoListByProcessId 49

Figura 27 - Estrutura do serviço validatePayment 50

Figura 28 – Default.view 51

Figura 29 – Ecrã desenvolvido 52
Figura 30 – Desenvolvimento do ecrã de pagamento 53
Figura 31 – Ecrã desenvolvido 54
Figura 32 – *Scheduled Task* 55
Figura 33 – Diagrama do processo de registo de entidade 60
Figura 34 – Modelo de dados do registo de entidade 61
Figura 35 – Processo de registo de entidade 62
Figura 36 – Documentos do registo de entidade 63
Figura 37 – Documento similarEntitiesList 64
Figura 38 – Documento entityRegistration 64
Figura 39 – Serviço checkIfEntityExistsInRegD 65
Figura 40 – *Input* 65
Figura 41 – *Output* 66
Figura 42 – Serviço manageUserCredencials 66
Figura 43 – *Input* 66
Figura 44 – *Output* 66
Figura 45 – Default.view 67
Figura 46 – Validation.view 68
Figura 47 – Ecrã criado 68
Figura 48 – Default.view 69
Figura 49 – Plataforma de pagamentos 72
Figura 50 – Módulo de transferência bancária 73
Figura 51 – Ecrã de carregamento de ficheiro 74
Figura 52 – Ecrã de carregamento de ficheiro 78
Figura 53 – Processo de pagamento por cartão de crédito 79
Figura 54 – Troca de mensagens 79

Lista de Tabelas

Tabela 1 – Descrição dos elementos da arquitectur 15
Tabela 2 – Passos importante do pedido de certidão 41
Tabela 3 – Exemplo de configuração 56
Tabela 4 – Passos importantes do registo de entidade 63
Tabela 5 - Lista de configurações para transferência bancária 76
Tabela 6 – Descrição do conteúdo do ficheiro 77
Tabela 7 – Configuração do módulo de multibanco 78
Tabela 8 – Configuração do módulo de cartão de crédito 81

Lista de Abreviaturas

AJAX – *Asynchronous JavaScript and Extensible Markup Language*

BO – *Back Office*

BPM – *Business Process Management* (Gestão de Processos de Negócio)

CAF – *Compposite Application Framework*

ESB – *Enterprise Service Bus*

FO – *Front Office*

Apache DirectoryServer – Apache DS

HTML – *HyperText Markup Language*

IS – *Integration Server*

JDBC – *Java Database Connectivity*

JSF – *Java Server Faces*

MIDAS – Método *Indra* para Desenvolvimento, Adaptação e Serviços

MIDO – *Método Indra Desenvolvimento*

MIND – *Método Indra Necessidades*

MITO – *Método Indra Transformação*

MISO – *Método Indra Serviços*

MVC – *Model-View-Control*

NIF – Número de Identificação Fiscal

PL/SQL – *Procedure Language*

RepD – Repositório de dispositivos

RegD – Registo de dispositivos

SOA – *Service-oriented Architecture* (Arquitectura Orientada a Serviços)

SIBS – Sociedade Interbancária de Serviços

TPA – Terminais de Pagamento Automático

TI – Tecnologias de Informação

URL – Uniform Resource Locator

XML – *Extensible Markup Language*

Capítulo 1 Introdução

Este relatório é escrito como parte integrante da unidade curricular Projecto de Estágio em Engenharia Informática, disciplina do 2º ciclo do Mestrado em Engenharia Informática, tendo a duração de nove meses.

O estágio foi realizado na *Indra Sistemas de Portugal*, tendo como orientação académica o Prof. Doutor João Pedro Guerreiro Neto docente no Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa e como coorientador da *Indra Sistemas de Portugal* o Engenheiro José Pedro Cardoso.

1.1 Contexto empresarial – *Indra*

A *Indra* é a maior empresa de Tecnologias de Informação (TI) em Espanha, e uma das principais multinacionais a nível europeu em TI. Actualmente tem escritórios em mais de 30 países e projectos em mais de 100 países, contando com mais de 35000 colaboradores internos.

A *Indra* está dividida em sete sectores principais: Administração Pública e Saúde, Transporte e Tráfego, Serviços Financeiros, Segurança e Defesa, Telecomunicações e Media, Energia e Indústria. Cada um destes sectores têm como suporte quatro áreas horizontais que se enquadram para servir o cliente, conforme Figura 1.

A constituição da *Indra Sistemas de Portugal* foi feita em 2003. Em 2007 houve a incorporação da *Azertia* e da *Soluziona*, também empresas de TI. Actualmente a *Indra Sistemas de Portugal* conta com mais de 400 profissionais.

Este projecto integra-se na área de *Soluções Tecnológicas* pertencendo ao sector da Administração Pública e Saúde. O projecto está a ser desenvolvido no cliente. O cliente é uma autoridade competente, que tem como atribuições a regulação, avaliação,

autorização, disciplina, inspecção e controlo de produção, distribuição, comercialização e utilização de alguns dispositivos existentes no mercado.

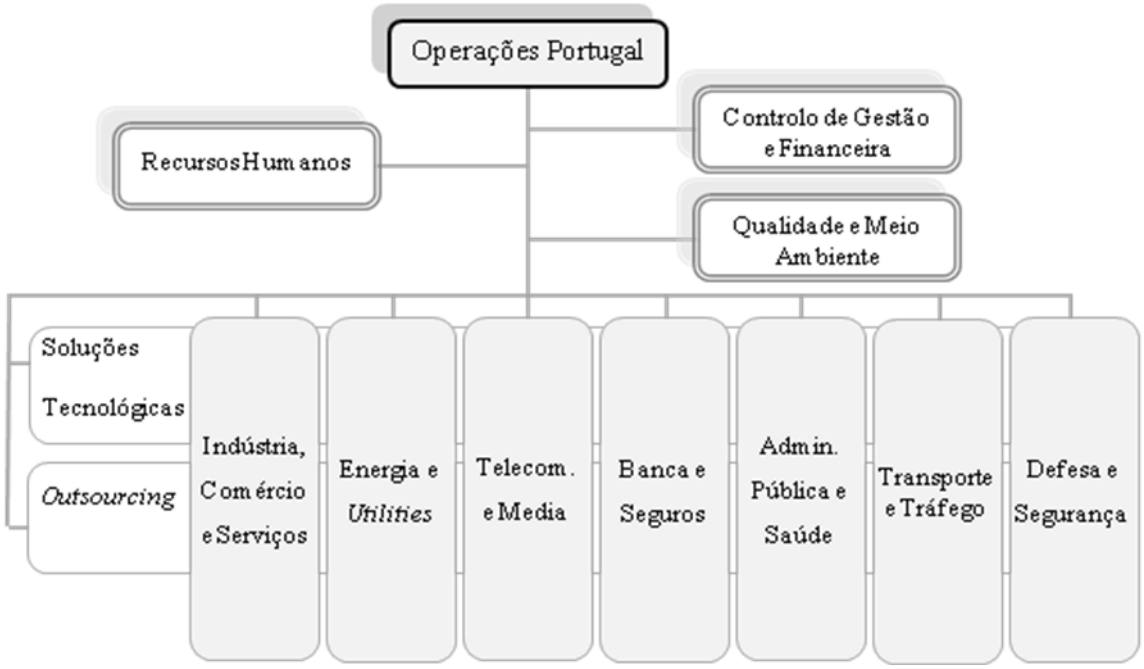


Figura 1 – Organigrama da *Indra*

1.2 Integração

O meu estágio na *Indra* deu início no dia 6 de Setembro de 2010.

Na primeira semana de estágio foi realizada uma sessão de acolhimento de novos elementos na *Indra*. Essa sessão teve a duração de cinco dias e, durante a mesma, foi-nos apresentada a origem da empresa e a estrutura *Indra*. Tivemos também uma sessão de acolhimento com os directores de cada área da *Indra*.

Tivemos igualmente diversas formações desde comunicação, serviço ao cliente e trabalho em equipa, onde, com várias actividades, se testava as nossas capacidades em determinadas áreas. Esta sessão de acolhimento foi um projecto pioneiro na *Indra* para a captação e integração de colaboradores juniores. A este projecto foi dado o nome de *Integrate*.

Após esta semana tive o início da integração no projecto onde realizei o meu estágio. Inicialmente realizei alguns tutoriais em *WebMethods* e posteriormente iniciei o meu projecto de estágio onde tive o apoio de toda a equipa do projecto, adquirindo assim numerosos conhecimentos na plataforma *WebMethods*.

1.3 Motivação e apresentação do problema

Este projecto nasce da necessidade de existir um processo global de suporte ao registo de dispositivos (RegD) por parte do cliente. Tem como principal foco a melhoria e automatização dos vários processos inerentes ao RegD, a estruturação da informação para exploração e consulta e a facilidade de comunicação entre o cliente e as entidades (fabricantes) dos dispositivos.

Este projecto traz como principal benefício um melhor serviço ao requerente e o desenvolvimento de futuras prestações públicas.

A infra-estrutura adoptada para este projecto foi construída com base nos produtos *WebMethods*, oferecendo produtos *end-to-end*. Uma mais-valia da ferramenta WebMethods é permitir a integração com vários sistemas legacy. A ferramenta Webmethods com base numa tecnologia SOA permite integrar diferentes tecnologias de arquitecturas díspares. Os produtos *WebMethods* permitem o desenvolvimento de processos e serviços ou a sua reutilização com características para sofisticadas soluções que permitem a sua rápida criação, implementação e gestão. Estes processos ou serviços podem ter como base outro sistema recorrendo a integração. No capítulo 2 e 4 é explicado com mais detalhe todo este processo de desenvolvimento.

1.4 Objectivos

O projecto tem como objectivo principal o registo de dispositivos, assim como as funcionalidades necessárias para a gestão dos vários processos consequentes ao dispositivo. Este registo sobre o dispositivo é pedido pelas entidades dos dispositivos e é executado pelo cliente. A gestão do dispositivo é realizada tanto à entidade como ao cliente.

Antes de começar o desenvolvimento deste projecto, houve a necessidade de compreender a delimitação do processo de negócio do cliente de modo a definir os objectivos do projecto e as aspirações do cliente assim como os prazos para o desenvolvimento.

No final de algumas reuniões com o cliente demarcaram-se os vários processos a serem desenvolvidos que iriam constituir o projecto:

- ✓ O registo de entidades;

- ✓ O pedido de elementos;
- ✓ Processo de pedido de cancelamento;
- ✓ A avaliação do registo;
- ✓ A alteração de dados;
- ✓ O pedido de certidão;
- ✓ O registo de dispositivos.

Para além dos processos definidos também se delimitaram algumas funcionalidades a serem desenvolvidas para o projecto, de modo a dar alguma versatilidade e moldar o projecto às necessidades do cliente:

- ✓ Gestão de utilizadores;
- ✓ Gestão de alertas;
- ✓ Gestão de conteúdos;
- ✓ *HelpOnline* para ajuda na navegação do RegD.

Cada um destes processos pretende especificar a sequência de passos e tarefas assim como os respectivos intervenientes e responsáveis por essas mesmas tarefas. Os intervenientes e responsáveis são entidades como o cliente ou a entidade, os quais podem completar tarefas ou despoletar um novo processo.

1.5 Organização do documento

Este relatório é constituído por cinco capítulos. Esta secção apresenta um pequeno resumo dos capítulos seguintes.

O segundo capítulo foca o planeamento técnico tendo em vista a implementação do projecto, abordando o processo de desenvolvimento, as camadas lógicas, a arquitectura do sistema e integrações que serão realizadas com sistemas externos.

No terceiro capítulo é apresentada a metodologia e planeamento do projecto, a metodologia *Indra*, o trabalho desenvolvido no projecto e o que falta concluir no mesmo. Também são abordadas as fases de execução do estágio comparando o plano previsto com o realizado.

No quarto capítulo é descrito o trabalho realizado, os processos desenvolvidos durante o estágio, a importância e impacto deste trabalho no projecto e a integração do trabalho desenvolvido com outros sistemas.

No quinto capítulo é apresentada a conclusão do trabalho e o trabalho futuro a ser desenvolvido no projecto e na *Indra*.

Capítulo 2 Planeamento Técnico

Neste capítulo serão descritos os detalhes técnicos do projecto efectuado com vista a implementação do RegD. Este planeamento é de extrema relevância pois serve para a orientação da equipa de trabalho e para manutenção futura.

Aqui estão documentadas as principais ferramentas usadas para o desenvolvimento do projecto, as componentes técnicas do processo, as entidades e serviços criados, as integrações com outros sistemas necessários e a estrutura e o fluxo de processos.

2.1 Ferramenta *WebMethods*

A plataforma *WebMethods* foi adquirida pela empresa *Software AG* em 2007. É uma plataforma especializada em integração de processos de negócio para empresas. A plataforma *WebMethods* é uma ferramenta de integração que inclui uma Arquitectura Orientada a Serviços (*Service-Oriented Architecture*, SOA) e a Gestão de Processos de Negócio (*Business Process Management*, BPM). Com esta ferramenta de integração podemos tirar partido de outras tecnologias que já estejam implementadas em empresas e desenvolver processos de *Web-Services* ou recorrer a *adapters* de modo a colmatar falhas ou necessidades das empresas. Assim, pode haver uma optimização dos vários processos de negócio, não sendo necessária uma modificação estrutural nesses processos.

Os principais produtos que a *suite WebMethods* oferece, e que foram utilizados no projecto, são os seguintes:

- ✓ *WebMethods BPM*:
  - o *MyWebMethods Server*;
  - o *WebMethods Compposite Application Framework (CAF).*
- ✓ *WebMethods Enterprise Service Bus (ESB):*
  - o *WebMethods Broker Server;*
  - o *WebMethods Integration Server;*

- *Adapters.*

2.1.1 *WebMethods Designer*

A ferramenta *WebMethods Designer* (imagem da ferramenta em) é uma ferramenta baseada em *Eclipse* que permite a modelação de processos, desenvolvimento gráfico de interfaces e criação de tarefas humanas recorrendo a tecnologia CAF, uma extensão da tecnologia *Java Server Faces* (JSF).

O *JSF* é uma *framework Model - View - Control* (MVC) para o desenvolvimento de aplicações *web* de forma visual, permitindo criar componentes visuais:

O modelo MVC é um padrão de arquitectura de software que visa separar a lógica de negócio da lógica de apresentação

O designer mostra toda a informação da aplicação usada na interface (*Bindings View*). Essa informação é gerada automaticamente ou manualmente a partir dos *beans*. As componentes JSF que se adicionam à interface podem estar interligadas a serviços. Essa ligação pode-se fazer de várias formas:

- Arrastar o conteúdo da *Binding View* para a componente JSF e o *Designer* automaticamente faz a interligação;
- Através da *Properties View* do JSF, aceder à ajuda (*Expression Binding Wizard*) onde se define a informação a interligar;
- Configurar a ligação manualmente;

No *Designer* é mostrado um editor do JSF. Está disponível uma *view* onde são disponibilizadas várias componentes (componentes JSF) que se podem usar na interface. Para usar uma componente basta arrastar essa componente JSF: o *Designer* automaticamente cria o *Extensible Markup Language* (XML) correspondente a essa componente e insere a componente na árvore da interface. As interfaces são guardadas num ficheiro .view. Em tempo de execução, o motor JSF da ferramenta *WebMethods* CAF interpreta o XML (que se encontra no ficheiro .view) para uma árvore de componentes JSF, e renderiza as componentes durante os pedidos efectuados passando o XML para *HyperText Markup Language* (HTML).

Quando se cria uma aplicação em CAF, o *Designer* automaticamente cria os descritores da aplicação, aceitando o *input* do utilizador originando uma acção baseada nesse *input.* Também se pode adicionar acções manualmente alterando ou adicionando

código JAVA directamente. Alguns dos controlos CAF recorrem à tecnologia *Asynchronous JavaScript and XML* (AJAX), que permite uma interacção assíncrona com o servidor.

Além do desenvolvimento dos ecrãs e dos processos, o *deploy* para o *MyWebMethods Server* pode ser feito através do *WebMethods Designer*.

O *MyWebMethods Server* é um portal server baseado em *Jetty* onde a aplicação é executada. Disponibiliza uma interface de administração que permite a gestão das componentes do próprio servidor, como por exemplo a forma de apresentação e o funcionamento dessas componentes, as permissões de acesso de utilizadores e grupos (*roles*).

2.1.2 *WebMethods Developer*

O *WebMethods Developer* (imagem da ferramenta em anexo) é uma ferramenta gráfica que permite desenvolver a integração lógica com os outros sistemas. Disponibiliza um ambiente de desenvolvimento que possibilita a criação, reutilização e o teste de serviços que constituem a solução de integração. O *Developer* permite a construção rápida da solução usando a linguagem chamada *WebMethods flow language*. A *flow language* é uma programação visual e fornece um conjunto de construções, criando acções a executar em tempo de execução. O *Developer* também permite a transformação da informação passada a cada acção. Essa informação é denominada de *pipeline* e a cada acção executada a informação da *pipeline* pode ser modificada dependendo das necessidades para a execução da próxima acção.

O *Integration Server* (IS) disponibiliza uma interface de administração que permite a configuração de várias componentes que fazem parte do próprio servidor. Essas componentes podem ser os pacotes a serem utilizados onde são guardados *Web-Services*, *adapters*, documentos e outras componentes, *schedulers* (procedimentos ou *Web-Services* que são executados num espaço de tempo definido), conexões (utilizadas pelos *adapters*), configurações de constantes, ligações a sistemas externos, entre outros.

2.2 Camadas lógicas

Neste projecto a aplicação irá assentar em três camadas lógicas. Cada uma destas será desenvolvida num componente próprio, a saber:

- ✓ Camada de apresentação, desenvolvida em *CAF;*
- ✓ Camada de lógica de negócio, desenvolvida em *flow*, no IS;
- ✓ Camada de dados, desenvolvida em *Oracle*.

A comunicação entre as camadas é feita apenas entre as camadas adjacentes. Dando um exemplo de modo a tornar perceptível a interacção entre as camadas, a um utilizador que aceda à aplicação usando o seu *Web Browser*, sabemos que a interacção deste com a aplicação dá-se através da camada de apresentação. A camada de apresentação, para compor os ecrãs de modo a apresentar ao utilizador, recolhe dados disponibilizados pela camada de lógica de negócio que, para disponibilizar esses mesmos dados, comunica com a camada de dados através de *Java Database Connectivity* (JDBC), e quando se revela necessário, com outros sistemas, por exemplo através de *WebServices*. A camada de dados disponibiliza uma interface escrita em pacotes, que encapsulam a lógica relacional dos dados que compõem a aplicação, facilitando desta forma evoluções e alterações – conforme Figura 2.

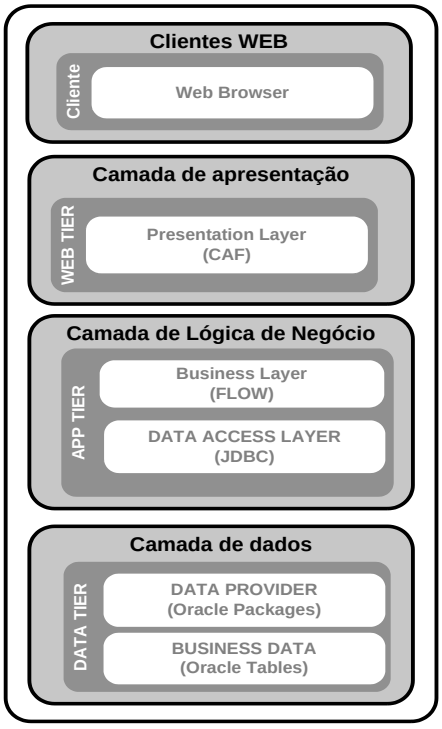


Figura 2 – Camadas Lógicas

Nos pontos seguintes apresenta-se a estratégia de desenvolvimento seguida para cada uma das camadas referidas na Figura 2.

2.2.1 Camada de apresentação

A camada de apresentação, tal como foi referido, é desenvolvida com recurso à tecnologia CAF da *suite WebMethods*. Nos vários documentos funcionais do projecto são apresentados e descritos os processos de negócio, os ecrãs associados, as regras de negócio definidas bem como as integrações com outras aplicações. Cada um dos processos, e dos ecrãs identificados nesses documentos, são desenhados com recurso aos componentes disponibilizados nesta tecnologia, utilizando a ferramenta de desenho de interfaces da própria *suite* (*Designer*). Para a interligação com a camada de lógica de negócio, são utilizados conectores a serviços escritos pela própria infra-estrutura (descrição mais pormenorizada no capítulo 4.1 deste documento).

O desenvolvimento de todas as *portlets* será feito num único projecto, sendo os diferentes tipos de acesso garantidos através de regras de segurança. As regras de segurança definem os parâmetros de acesso a determinados conteúdos da aplicação. Por exemplo, se um ecrã de pesquisa é utilizado entre o *BackOffice* (BO) e o *FrontOffice* (FO) as funcionalidades que deverão estar disponíveis apenas em BO serão apresentadas a quem disponha de uma regra de segurança com o nome *RegDBackOffice*.

2.2.2 *Portlets*

Para melhor compreensão e orientação durante o desenvolvimento do projecto foram definidas padronizações que serão agora descritas.

Todas as *portlets* para utilização comum têm o nome no formato *Common<<Nome sugestivo>>*. Se uma *portlet* é específica de BO ou FO, o seu nome começará por *BackOffice* ou *FrontOffice*, respectivamente. Para que seja possível no futuro dar suporte a uma *interface multilingue*, todas as *labels* que compõem a *portlet*, deverão ser escritos no ficheiro de *resources*, havendo uma associação do texto das *labels* a cada recurso.

O formato deste ficheiro de texto é o do ficheiro de tipo *properties* do JAVA. Esta forma de suporte, a multilingue, é o standard para aplicações JSF e portanto, também para CAF. Do ponto de vista de quem traduz, a única operação necessária será disponibilizar um ficheiro de *properties* para o *locale* certo mantendo o nome das propriedades, e a infra-estrutura encarregar-se-á de apresentar as páginas para o *locale*

certo para cada utilizador. De seguida é mostrado um exemplo do ficheiro de *resources* para a língua Portuguesa, "*common.label.valueToSearch=Valor a Pesquisar*".

Para além da composição do aspecto gráfico da *portlet*, é importante referir que as associações dos controlos que o utilizador utiliza para inserir os valores na página são feitos através de um *managed bean* criado para o efeito.

De forma a concentrar todos os *WebService Connectors* num único ponto, todas as *portlets* contêm um *bean* com o nome *<<Nome Sugestivo>>Services*, por exemplo *DeviceSearchServices*. À semelhança desta organização, também é criado um *bean* com o nome *<<Nome Sugestivo>>Providers*, onde são adicionados todos os *providers* necessários para compor a interface gráfica. Tanto um como outro *bean* derivam de *com.webmethods.caf.faces.bean.BaseFacesSessionBean*, dado que vão ser usados apenas como entidade agrupadora de outros *beans*.

2.2.3 Camada de lógica de negócio

A camada de lógica de negócio é desenvolvida em *flow*. Estes serviços são desenvolvidos num pacote próprio, com a estrutura descrita na Figura 3.

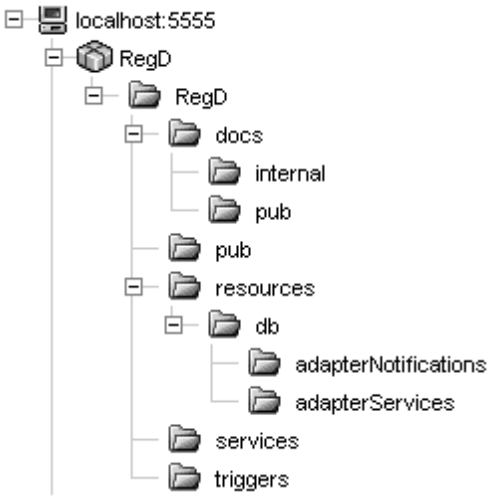


Figura 3 – Estrutura do pacote no IS

Cada uma das pastas que compõem o pacote contém os elementos que se encontram listados nos pontos seguintes:

- ✓ *Docs/Internal* – Aqui residem os documentos que não são publicáveis;
- ✓ *Docs/Pub* – Os documentos publicáveis estão contidos nesta pasta;

- ✓ *Pub* – Nesta pasta deve ser adicionado uma pasta por cada funcionalidade ou elemento. Nesse elemento estão contidos os serviços a serem disponibilizados à camada de apresentação desenvolvida em CAF conforme já foi descrito. É garantido que nenhum serviço a utilizar pela camada de apresentação está fora desta estrutura;
- ✓ *Resources/db/adapterNotifications* – Todos os *adapterNotifications* necessários no processo de desenvolvimento estão nesta pasta;
- ✓ *Resources/db/adapterServices* – Os *adapter services* necessários estão nesta pasta e com estes conseguimos fazer conexões à base de dados;
- ✓ *Services* – Os serviços desta pasta são todos aqueles que não são públicos;
- ✓ *Triggers* – Como o nome indica, os *trigger* necessários à aplicação residem nesta pasta.

Para além deste pacote, existe um outro, com o nome *REGDProcesses* onde estão os serviços necessários para cada um dos processos. Nesse pacote, os serviços estão agrupados em pastas em que cada um representa um processo.

2.2.4 Camada de dados

A interacção da camada de negócio com os dados é feita através de pacotes escritos em *PL/SQL* desenvolvidos para o efeito.

Por cada funcionalidade existe um pacote que disponibiliza a interface necessária à funcionalidade em causa. Esta aproximação permite um melhor encapsulamento dos dados, e uma maior facilidade de evolução/adaptação, para além de dividir os serviços da lógica relacional.

2.2.5 Arquitectura tecnológica

As camadas lógicas descritas anteriormente assentam, em termos de arquitectura, nos serviços disponibilizados pela *suite WebMethods*. Tal como foi descrito, a interacção com o utilizador é feita através do *MyWebmethodsServer*, que por sua vez comunica com o IS para compor a interface gráfica a apresentar ao utilizador. A lógica desenvolvida no IS comunica com a base de dados através de JDBC e com outros sistemas através de serviços ou *WebServices* que estejam disponíveis. O modelo de persistência da aplicação é suportado num esquema de base de dados relacional *Oracle*.

Para além dos serviços já enunciados, é também utilizado um repositório de utilizadores exterior à *suite WebMethods*, tanto para os utilizadores internos como externos, esses utilizadores são guardados no *Apache Directory Server* (Apache DS).

Apresenta-se na Figura 4 os elementos que compõem a arquitectura tecnológica que suporta a aplicação.

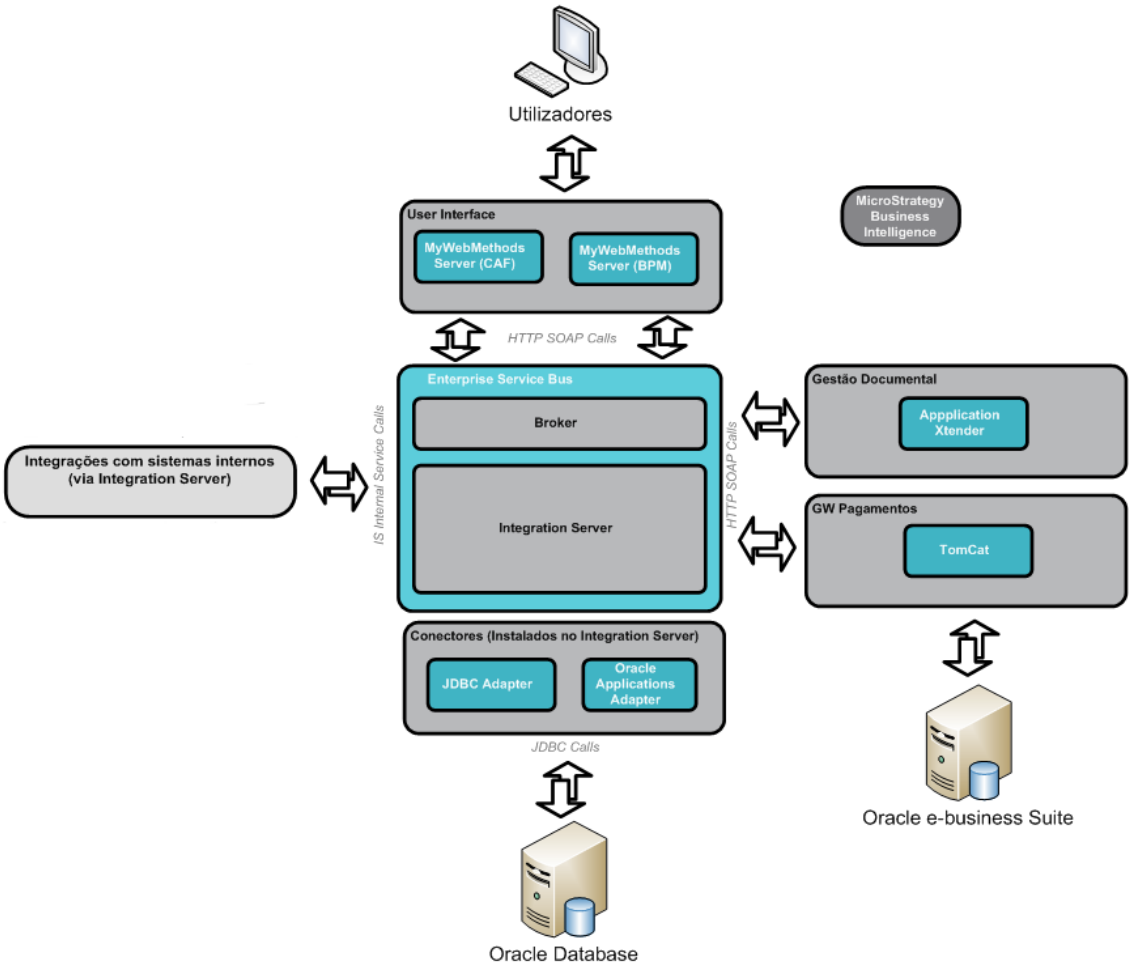


Figura 4 – Arquitectura tecnológica

De seguida é apresentada uma tabela com o sumário dos elementos da arquitectura de modo a compreender melhor a sua função.

Elemento	Descrição
Clientes	Os clientes da aplicação utilizam *WebBrowsers* para aceder à aplicação, sendo suportados aqueles que se encontram descritos na literatura da suite *WebMethods*, versão 7.1.2 .

MyWebMethodsServer	*MyWebMethodsServer*, responsável pela apresentação da interface gráfica da aplicação.
Broker	Serviço de distribuição de mensagens, utilizado maioritariamente para o lançamento e subscrição de eventos.
IS	É no IS que se encontra residente a lógica de negócio, sendo também aqui que se processam os eventos referidos atrás.
Oracle	Base de dados de suporte à aplicação.
Apache DS	Repositório de utilizadores externos, sistema interno.
Reference Data	Referências de dados que devem ser utilizados para popular as listas comuns entre projectos.
Active Directory	Repositório de utilizadores, sistema interno.
Gestão de entidades	Repositório de entidades, sistema interno.
***Gateway* de Pagamentos**	Servidor aplicacional onde se registam e efectuam pagamentos.

Tabela 1 – Descrição dos elementos da arquitectur

2.2.6 Arquitectura física

A arquitectura física foi desenvolvida com base nos recursos disponibilizados pelo cliente. Como a tecnologia a utilizar é *WebMethods* o cliente já possuía uma infra-estrutura suficiente para dar suporte a aplicação em ambiente de produção e testes. Na Figura 5 está representado uma estrutura física similar à do cliente, não sendo a que realmente está em uso nas instalações.

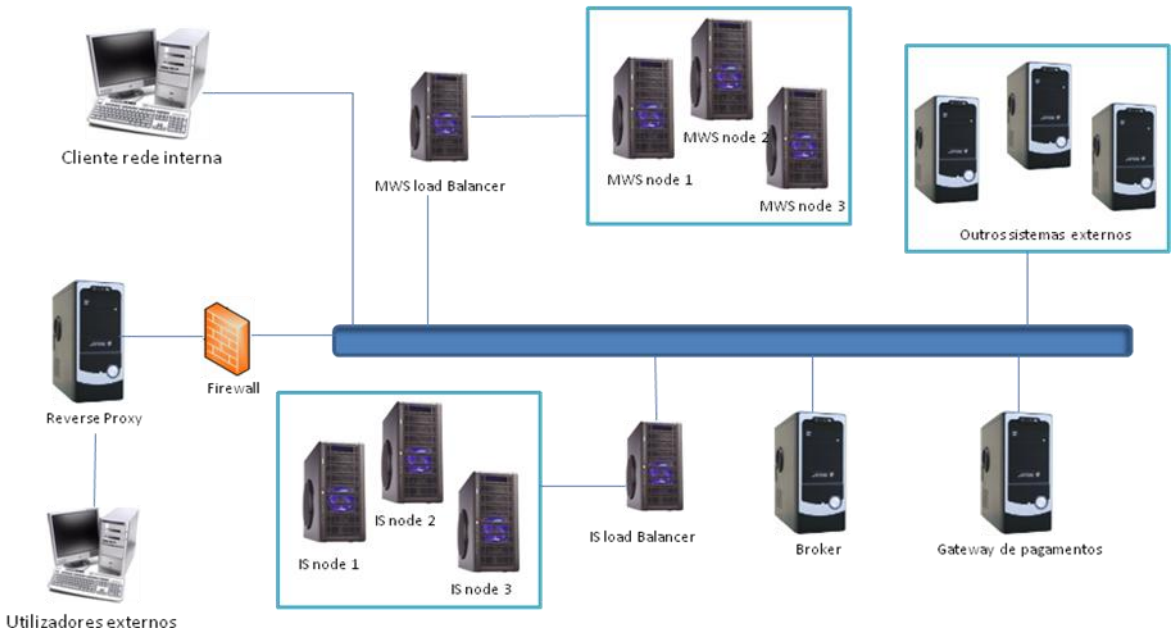


Figura 5 – Infraestrutura física

Quando o utilizador externo aceder à aplicação é efectuado um pedido ao *Reverse Proxy*, o qual se encarrega de fazer um pedido ao *load Balancer*. Para os utilizadores da rede interna os pedidos serão direccionados automaticamente para o próprio *load Balancer*. O *load Balancer* tem a função de distribuir a carga pelos servidores mantendo a integridade de todos os dados.

Devido à dimensão do projecto (e também futura utilização) houve a necessidade de recorrer a dois balanceadores de carga, um encarregue de balancear os pedidos para o servidor *MyWebMethodsServer* e outro encarregue de balancear os pedidos do *IS*. Os utilizadores, apesar de não acederem directamente ao *IS*, interagem directamente com o *MyWebMethodsServer*. O *MyWebMethodsServer* utiliza os serviços que se encontram no *IS*, o que isso origina um número elevado de pedidos ao servidor.

2.3 Modelo de dados

2.3.1 Integraçao do modelo de dados

O modelo de dados que suporta a aplicação foi desenhado num projecto desenvolvido anteriormente e os elementos centrais da actual aplicação são os mesmos que se encontravam nesse projecto.

Durante a fase de análise funcional para o modelo de dados, foram verificadas algumas diferenças entre os conteúdos que são comuns aos dois projectos, logo, o modelo de dados sofrerá (ainda não foi feita esta integração) algumas alterações para suportar as referidas diferenças. Além dessas alterações, o modelo deverá sofrer mais alterações na altura da implementação dos processos da aplicação. Tanto as alterações a fazer ao modelo de dados, como as novas tabelas a acrescentar, serão feitas sobre o actual modelo de dados do projecto já desenvolvido anteriormente.

2.3.2 Modelo de dados RegD

Para que exista uma interligação entre os processos suportados pela *suite WebMethods* e a base de dados do RegD, existe no modelos de dados um conjunto de tabelas alimentadas através dos processos de negócio, e nas quais se gravam os dados, tanto dos processos como das tarefas realizadas.

A forma de gravar os dados dos processos passará pela inclusão de passos nos processos BPM que permitem a recolha destes dados. Para as tarefas serão incluídos eventos nas acções de alteração e fecho que permitiram a recolha dos dados necessários. Por questões de facilidade de suporte, existe uma correlação entre os dados de negócio e os da infra-estrutura.

Quanto ao modelo de dados, que representa o histórico das alterações feitas ao esquema principal, interessa referir que a sua implementação passará pela replicação das tabelas que constituem o esquema principal.

2.4 Serviços do IS

O conjunto de serviços desenvolvidos, para utilização nas páginas, encontra-se no pacote *DeviceManagementPackage*. Os serviços utilizados nos processos encontram-se no pacote *DeviceManagementProcesses*.

A documentação dos serviços pode ser obtida em tempo real. Para a obtenção da documentação de todos os serviços de um pacote, pode ser consultado o endereço http://<IS Server>:5555/packageName=<nome do package>. Esta informação é acedida apenas por utilizadores internos do cliente e encontra-se disponível na página de administração do *IS*.

2.5 Integrações a serem desenvolvidas com sistemas externos

A integração com outros sistemas será feita sempre através do IS, quer por serviços disponibilizados por outros projectos, quer por *WebServices* que se encontram disponíveis. Nos pontos seguintes faz-se uma breve descrição da integração com os sistemas que identificámos como necessários para este desenvolvimento.

2.5.1 *Reference data*

Neste sistema existem os dados utilizados para popular as listas que são comuns entre projectos, por exemplo, listas de países.

A integração com este sistema é feita através de serviços IS.

2.5.2 Repositório de entidades

Todas as entidades que interagem com o cliente têm os seus dados genéricos guardados no repositório deste projecto. A interacção com este sistema é feita por duas vias.

Para guardar ou alterar dados, são utilizados serviços disponíveis no IS.

Para o desenvolvimento de pesquisas que disponibilizem dados genéricos e também específicos, é utilizado o pacote existente para o efeito..

2.5.3 Gestão documental

No que respeita à integração com a gestão documental, a integração que é necessária prende-se com a inserção de novos documentos e a disponibilização de documentos ao utilizador.

2.5.4 *Gateway* de pagamentos

O núcleo da *gateway* de pagamentos é um sistema transversal à organização, podendo ser usado por outros sistemas. Os sistemas que possuam processos de negócio específicos para pagamentos recorrem à *gateway* de pagamentos de forma a poder completar o seu ciclo de vida.

O sistema de *gateway* de pagamentos é responsável pela gestão dos pagamentos a efectuar e pela validação dos pagamentos que já foram efectuados. Para cumprir este

último objectivo a *gateway* de pagamentos permite não só carregar a informação proveniente de instituições financeiras/bancárias, como também registar o próprio acto do pagamento, podendo também haver a consulta desses pagamentos.

A *gateway* de pagamentos tem uma parte relevante para o desenvolvimento deste relatório porque o trabalho desenvolvido interage com a *gateway* de pagamentos.

Adiante será explicado em que contexto é necessária a utilização da *gateway* e o impacto no trabalho desenvolvido.

Capítulo 3 Metodologias e planeamento

Este capítulo aborda a metodologia adoptada e o planeamento do projecto. Inclui uma descrição esquemática do trabalho que fora realizado quando integrei o projecto bem como dos objectivos cumpridos neste estágio.

Para o projecto de estágio foi-me incumbido o desenvolvimento da segunda fase do projecto, designada por *Milestone 2* e alguns processos da Fase 1, designada por *Milestone 1*.

Inicialmente darei foco à metodologia *Indra* assim como a forma de organização e implementação dessa metodologia.

3.1 Método Indra para Desenvolvimento, Adaptação e Serviços - MIDAS

Para o desenvolvimento de projectos a *Indra* possui várias ferramentas corporativas de apoio aos projectos em desenvolvimento.

Toda a documentação envolvida no projecto é gerida por uma ferramenta designada Método *Indra* para Desenvolvimento, Adaptação e Serviços (MIDAS). Esta ferramenta é um guia para o desenvolvimento dos projectos e é particionada em quatro etapas, conforme Figura 6:

- ✓ *Método Indra Necessidades* (MIND) – Nesta etapa é realizado um estudo inicial do problema e o enfoque da solução. É definida a solução técnica nesta fase;
- ✓ *Método Indra Desenvolvimento* (MIDO) – Anotam-se a aquisição, adaptação ou reutilização de várias componentes de outros projectos para a execução do novo projecto assim como todo o desenho e análise do projecto;
- ✓ *Método Indra Transformação* (MITO) – Definem-se os parâmetros de integração do novo projecto com os sistemas já existentes. Aqui também é guardada a validação do projecto e a entrega do projecto;

- ✓ *Método Indra Serviços* (MISO) – Desenvolvem-se o plano de garantia e manutenção do projecto.

Figura 6 – Estrutura MIDAS

3.2 Planeamento do projecto

Devido à dimensão do projecto e dos processos nele implícitos foram definidas quatro fases de projecto (*Milestones*). Em cada uma são desenvolvidos vários processos. Após cada fase é feita uma entrega do projecto ao cliente de modo a averiguar a sua adequação.

De seguida estão descritas os processos principais de cada uma das quatro *Milestones*.

- ✓ *Milestone* 1:
  - Processo de registo de dispositivo;
  - Processo de pedido de elementos;
  - Processo de registo de entidade;
  - Processo de avaliação do dispositivo;
  - Processo de alteração do dispositivo.
  - …
- ✓ *Milestone* 2*:*
  - Processo de pedido de certidão.
- ✓ *Milestone* 3*:*
  - Processos de gestão do cliente
- ✓ *Milestone* 4*:*
  - Indicadores.

Para cada processo das quatro fases são elaborados documentos com o levantamento de requisitos, a análise funcional, os planos de testes e também as actas das reuniões onde são descritos os tópicos principais de cada reunião. Todos estes documentos terão de ser aprovados pelo cliente.

3.3 Recursos

Para desenvolver este projecto foi destacada uma equipa, organizada em duas sub-equipas: uma funcional e uma técnica.

A equipa funcional realiza reuniões com o cliente com o intuito de identificar os requisitos funcionais e não funcionais de cada processo bem como o fluxo do mesmo. Tendo por base as reuniões realizadas, são elaborados os cadernos de requisitos e de especificação funcional para cada processo, bem como os casos de testes.

A equipa técnica efectua a implementação do projecto, ou seja, a criação e desenvolvimento do processo baseada nos documentos gerados pela equipa funcional. A equipa técnica interliga todos os componentes descritos no planeamento técnico. Por fim, após a identificação dos erros do projecto, cabe à equipa técnica a correcção desses mesmos erros.

Tendo em conta o projecto de estágio e por causa de limitações do projecto, a minha integração no projecto passou pelas duas equipas, inicialmente pertencendo à equipa técnica, implementando o processo baseado nos documentos e posteriormente integrando a equipa funcional fazendo o levantamento de requisitos e análise funcional de um outro processo planeado.

3.4 Trabalho realizado

O projecto, aquando a minha chegada, encontrava-se em fase de desenvolvimento da *Milestone* 1. Foram efectuados os levantamentos de requisitos e análise funcional e encontravam-se em fase de conclusão alguns processos.

Os processos em fase de conclusão são submetidos a testes. Em fase de testes são criados *tickets*. Caso existam, esses erros são reportados num *track* onde posteriormente são corrigidos. Posteriormente são feitos testes pelo cliente. O processo dá-se como concluído quando não houverem mais erros a serem reportados e quando todos os erros detectados forem concluídos.

Enquanto estão a ser desenvolvidos os processos, também são revistos os documentos de análise e requisitos dos processos em desenvolvimento de modo a colmatar algumas falhas ou acrescentar melhorias aos processos anteriormente descritos. Estas falhas ou melhorias podem ser detectadas pela equipa ou pelo cliente.

Em relação ao projecto de estágio, no momento da minha chegada, o projecto estava numa fase de grande desenvolvimento e pouca análise. Devido à necessidade de recursos para o desenvolvimento optou-se por começar o meu estágio na equipa técnica onde implementei o processo da *Milestone* 2, o pedido de certidão.

Houve uma fase de adaptação à *WebMethods* e uma aprendizagem com as ferramentas de trabalho.

A *Milestone* 2 já tinha a documentação gerada pela equipa funcional e pré-aprovada pelo cliente, após a aprovação final do cliente transitou para a fase de desenvolvimento.

Após a conclusão da fase de desenvolvimento da *Milestone* 2, passei para a fase funcional. O processo ao qual fui integrado ainda não tinha a análise funcional completa. O processo é o registo de entidade, parte integrante da *Milestone* 1 e após ter realizado a fase de análise do processo fiz a sua implementação completando assim o objectivo deste estágio.

Além do desenvolvimento destes processos, foi-me ainda atribuída a tarefa de gestão da *gateway* de pagamentos, um sistema transversal ao RegD e que é utilizado no pedido de certidão para realizar pagamentos.

Seguidamente encontra-se descrito o plano de trabalhos, durante os nove meses de estágio, com as etapas principais do projecto e uma pequena descrição sobre o trabalho desenvolvido.

3.4.1 Implementação da *Milestone* 2- pedido de certidão

O objectivo desta fase é efectuar a implementação do processo levantado na fase de análise funcional e requisitos. Esta fase inclui o desenho e navegação de ecrãs, as regras de negócio e os serviços necessários à execução do processo.

3.4.2 Implementação da *gateway* de pagamentos

Implementar a *gateway* de pagamentos no âmbito dos vários projectos desenvolvidos no cliente.

Nesta fase destaca-se a implementação de várias formas de pagamento: por valores, multibanco, transferência bancária e cartão de crédito. Além destas formas de pagamentos são também implementadas medidas de segurança de modo a disponibilizar pagamentos online através do cartão de crédito.

3.4.3 Análise funcional de requisitos da *Milestone* 1 – registo de entidade

Análise funcional, requisitos e serviços de suporte à execução do processo de registo de entidade. Definição dos requisitos de integração com a aplicação de *Gestão de entidades* (GestEnt) já existente no cliente.

3.4.4 Implementação da *Milestone* 1 – registo de entidade

Efectuar a implementação dos processos levantados na fase de análise funcional e requisitos.

Esta fase inclui o desenho do fluxo do processo e a descrição dos caminhos principais e de excepção do mesmo. Inclui também o desenho dos ecrãs associados, fluxo de navegação de ecrãs, regras de negócio e os serviços necessários à execução dos processos. Neste processo foi feita uma sincronização com outro projecto anteriormente desenvolvido, o repositório de entidades, e também a um *Apache Directory*. Posteriormente será descrito como é que esta sincronização será feita.

3.4.5 Validação e testes

Efectuar a validação funcional e técnica através da execução dos casos de testes definidos para cada um dos processos levantados na fase de análise funcional e de requisitos.

Os testes são realizados, numa primeira fase, pela equipa funcional da *Indra*, a qual vai reportando à equipa técnica as ocorrências detectadas. Quando obtêm uma percentagem, em termos de qualidade, superior a 85%, o processo é passado para o cliente. Inicia-se, neste momento, a fase de testes de aceitação realizada pelo cliente. Os casos de teste descritos nos cadernos são realizados tanto pela equipa da *Indra* como pela equipa do cliente.

3.4.6 Relatório final

O estágio terminou com a escrita e entrega do relatório final, onde é descrito e documentado o trabalho realizado.

Encontra-se em anexo no relatório um mapa de *Gantt* com a duração das tarefas que foram desenvolvidas ao longo do estágio.

Capítulo 4 Trabalho desenvolvido

Este capítulo descreve o trabalho realizado para a concretização do estágio. O trabalho consistiu no desenvolvimento de módulos para a aplicação RegD. Para o desenvolvimento deste trabalho foi utilizada a plataforma *WebMethods*, e como suporte de dados da aplicação, foi utilizada uma base de dados *Oracle*. A camada de apresentação foi desenvolvida com recurso à tecnologia CAF, incluída na *suite WebMethods*.

Inicialmente desenvolvi a *Milestone* 2 do projecto, o processo de pedido de certidão e deste modo adquiri conhecimentos da ferramenta contribuindo assim para uma melhor e mais rápida aprendizagem na tecnologia.

Este processo tem um factor importante na aplicação devido à integração que tem com aplicações externas ao sistema do RegD, nomeadamente da *gateway* de pagamentos.

Posteriormente efectuei a análise funcional e desenvolvi um processo da *Milestone* 1, o registo de entidade. Com este processo adquiri conhecimentos de análise e desenho do processo, inclusive do seu desenvolvimento.

Como estava envolvido no processo de pedido de certidão, fiquei encarregue de fazer a integração e configuração do sistema da *gateway* de pagamentos com os projectos desenvolvidos. A *gateway* de pagamentos tem como objectivo ser um sistema transversal a qualquer aplicação, através da qual se podem efectuar os pagamentos necessários a qualquer aplicação.

4.1 Desenvolvimento em *WebMethods*

4.1.1 Desenvolvimento do processo BPM

O processo BPM tem uma importância fundamental para a implementação futura de todos os serviços e tarefas.

Todo o trabalho desenvolvido durante a fase de análise tem grande influência nesta fase, pois é aqui que se define o ciclo de vida de cada processo. É fundamental ter em atenção o desenvolvimento do processo tendo em vista a sua optimização e a sua fácil manutenção e escalabilidade.

Para o planeamento inicial consideram-se duas características muito importantes: a atomicidade e a simplicidade do processo.

4.1.2 Desenvolvimento e implementação de serviços no *IS*

Todos os *WebServices* desenvolvidos em *WebMethods* têm o conceito de SOA, permitindo assim a reutilização de outros serviços já desenvolvidos, mesmo que esses serviços sejam de outra aplicação transversal ao RegD. Esta reutilização de serviços transversais à aplicação é uma das grandes vantagens da plataforma *WebMethods*. No desenvolvimento dos serviços tivemos em conta algumas características nomeadamente a sua atomicidade e a sua generalização.

Quanto mais complexa for a implementação de um serviço, menos reutilizável será diminuindo assim a sua atomicidade. A generalização de um *WebService* tem em vista a sua reutilização futura.

O mais importante no desenvolvimento de um *WebService* é a sua função, para o que se destina. Para isso são definidos os seus parâmetros de entrada e de saída, ou seja, o que queremos que o serviço devolva tendo em conta os parâmetros de entrada.

Como parâmetros de entrada/saída podem ser passados documentos ou variáveis. Além da organização da estrutura do serviço, o uso de documentos como parâmetro evita a substituição não intencionada de valores de alguma variável durante a execução do serviço. Isto acontece devido à existência de variáveis com o mesmo nome que ficam numa *pipeline* durante a execução de um serviço. Para evitar alguma substituição não

intencionada de valores durante algum serviço, convém eliminar essa variável ou documento da *pipeline*, para isso usa-se a acção *drop*. Assim evita-se que a pipeline contenha *lixo* ficando mais organizada.

Para o desenvolvimento dos serviços é possível aceder a uma *pipeline* onde é guardada a informação necessária para a execução do serviço. Os valores guardados na *pipeline* podem ser removidos, acrescentados ou modificados durante a execução dos vários passos do serviço.

Durante o desenvolvimento de serviços são constantes as transacções com a base de dados (estas transacções são disponibilizadas pela plataforma *WebMethods*).

Para fazer um pedido à base de dados, utilizam-se *adapters*. Estes *adapters* são desenvolvidos pela equipa de desenvolvimento e implementam a linguagem *Procedure Language* (PL/SQL).

4.1.3 Desenvolvimento e implementação das tarefas em CAF

Conforme já foi referido, a ferramenta *Designer* (baseada em J2EE) pertence à suite de *WebMethods* e permite o desenvolvimento de componentes gráficas através da tecnologia CAF. Aqui é desenvolvida a camada de apresentação que faz a interligação com a camada de lógica de negócio.

Alguns dos *WebServices* criados no IS são invocados durante o desenvolvimento das tarefas. Deste modo consegue-se abstrair a componente gráfica do desenvolvimento dos *WebServices*.

Se tivermos um *WebService* no IS que precisamos de utilizar na aplicação CAF. O Designer permite invocar o serviço fazendo as ligações automaticamente. Para isso basta apenas localizar o serviço e arrastar o serviço para a tela de design. O Designer irá criar um conector do *WebService*, com os seguintes itens descritos em baixo:

- ✓ O Designer cria os controlos para os inputs e outputs do serviço, usando componentes apropriados com base nos tipos de dados. Além disso o serviço adiciona um botão de refresh para invocar o serviço.

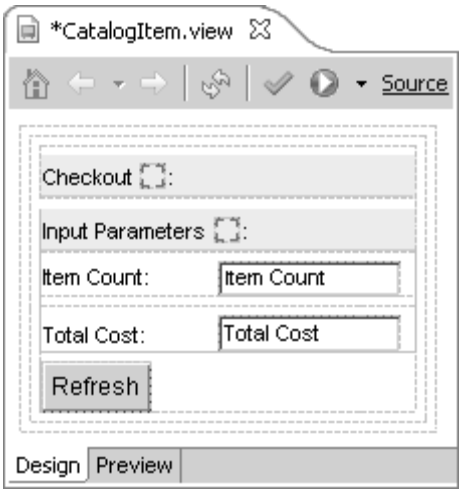


Figura 7- Input e Output do serviço

- ✓ Na *BindingView* (Figura 8) o designer adiciona um conector do *WebService* sobre o *page bean*. O elemento conector do *WebService* contem uma estrutura de elementos. Isso inclui propriedades que correspondem às entradas e saídas do serviço. O Designer acrescenta também:

    - o As propriedades que correspondem ao *Input/Output* (Figura 7) do serviço Estas propriedades são usadas para preencher dinamicamente o conteúdo usado para a saída e entrada dos serviços.

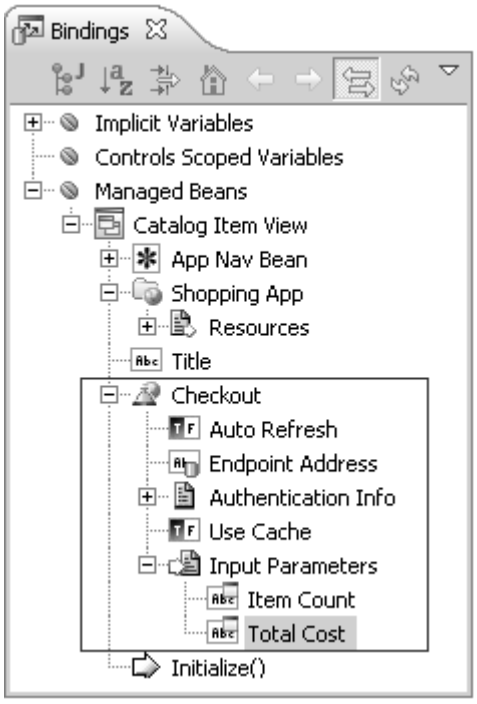


Figura 8- Conector com a estrutura do serviço

- Propriedades de configuração, tais como a informação de autenticação ou o *EndPoint* e *cache.*
- Um método *refresh()* que invoca o serviço.

4.2 Implementação do processo de pedido de certidão (*Assignment* 2)

Neste subcapítulo é descrita a implementação do processo de pedido de certidão. Como é um dos maiores processos da aplicação, o pedido de certidão é o único processo da *Assignment* 2.

O desenvolvimento engloba tanto a parte de FO, onde as entidades fazem a submissão do pedido de certidão e pagamentos, como a parte de BO, onde é realizado a avaliação e aceitação do dispositivo submetido para pedido.

Nos subcapítulos seguintes será definida a importância e o objectivo deste processo em toda a aplicação, assim como todos os intervenientes e integração realizada com os sistemas externos.

4.2.1 Âmbito e especificação do processo

Este processo tem como objectivo permitir que as entidades solicitem pedidos de certidão *on-line*. Anteriormente, este pedido era apenas efectuado através da submissão de pedidos em papel, o que fazia com que o processo fosse complicado e moroso.

O processo de pedido de certidão, assim como toda a aplicação, é desenvolvido em *flow*, pelo que, cada passo do processo está apenas acessível quando os passos antecedentes forem concluídos. Para ser mais perceptível pelo cliente, foi desenvolvido um esboço de todos os passos inerentes ao processo assim como uma descrição geral de cada passo do processo.

4.2.2 Desenho e descrição

Para efectuar um pedido de certidão tanto a entidade como o dispositivo terão que estar registados no portal.

Este processo pode ser despoletado pela entidade. Primeiramente a entidade tem que efectuar uma pesquisa dos seus dispositivos, seleccionar os dispositivos para os quais pretende fazer o pedido e activar o pedido de certidão.

Após efectuar o pedido, é feita uma validação pelo sistema. No caso dessa validação não ocorrer com sucesso, o sistema não despoleta o processo de pedido de certidão e apresenta uma mensagem ao utilizador com esta informação.

Após esta validação, e caso os dispositivos seleccionados sejam todos do mesmo tipo, é mostrada uma *pop-up* onde a entidade pode escolher como deseja receber a certidão, por correio ou ir buscá-la presencialmente. Após esta escolha, o estado do processo é actualizado para "Pendente para Pagamento Inicial" apenas para FO. Seguidamente são gerados os dados referentes ao pagamento sendo estes enviados à entidade para que efectue o pagamento do pedido de certidão de modo a que o processo avance. Estes dados do pagamento são gerados pela *gateway* de pagamentos, onde é permitido registar e efectuar os pagamentos. O pagamento inicial, quando é registado na *gateway*, tem o estado de "não_pago".

Em BO o processo não é despoletado enquanto não for realizado o pagamento. A entidade tem um período de 30 dias para efectuar o pagamento. Se após esse período a entidade não tiver efectuado o pagamento, o processo expira automaticamente, actualizando o estado do processo para "Cancelado" e alertando a *gateway* de pagamentos, passando a nota de pagamento para "Cancelado".

Se o pagamento tiver sido efectuado com sucesso, é iniciado o processo de pedido de certidão para gestor, actualizando o estado do processo. Em consequência também é gerada uma tarefa de avaliação dos dispositivos para os quais foi efectuado o pedido.

Na tarefa de avaliação o gestor avalia os diferentes dispositivos pertencentes ao pedido. Nesta tarefa será também é possível despoletar um pedido de elementos apenas para a entidade.

No caso de todos os dispositivos terem sido avaliados, o gestor dá como concluída a avaliação dos dispositivos. Após essa conclusão o sistema gera uma nova tarefa de análise de certidão onde o gestor poderá analisar o pedido de certidão e gerar certidões.

Após efectuada esta análise e a possível geração de certidão, o gestor conclui a tarefa de análise.

Independentemente da análise do gestor, seja ela aceite ou recusada, é gerada uma nova tarefa de avaliação de director. O director analisa o pedido feito pela entidade e analisa a avaliação e análise efectuada pelo gestor. Após esta análise o director pode revogar ou aceitar o pedido. Se o pedido for revogado o estado do processo irá mudar para "Cancelado" e a entidade será notificada. No caso de ser aceite o sistema irá calcular se existe pagamento adicional (com base no conteúdo da certidão). No caso de existir um pagamento adicional, o processo terá o mesmo comportamento aquando no pagamento inicial. Tal como acontece com o pagamento inicial, a entidade possui 30 dias para realizar o pagamento adicional. Se o pagamento for efectuado com sucesso, será gerada uma nova tarefa de emissão de certidão sendo também actualizado o estado do processo.

Por fim, é criada uma tarefa de emissão para o gestor, onde este poderá assinar e carimbar a certidão, fazendo *download* da certidão. Depois de feito o *upload* das actualizações da certidão, pode ser completada a tarefa dando assim o processo como "Concluído" terminando assim o ciclo do processo de pedido de certidão.

Ao ser completada esta tarefa é enviado um e-mail para a entidade com a informação de que a certidão já se encontra disponível.

Para uma melhor noção de todo o processo, de seguida apresenta-se um esquema ilustrativo do ciclo de vida do processo de pedido de certidão desenvolvido em BPM, Figura 9.

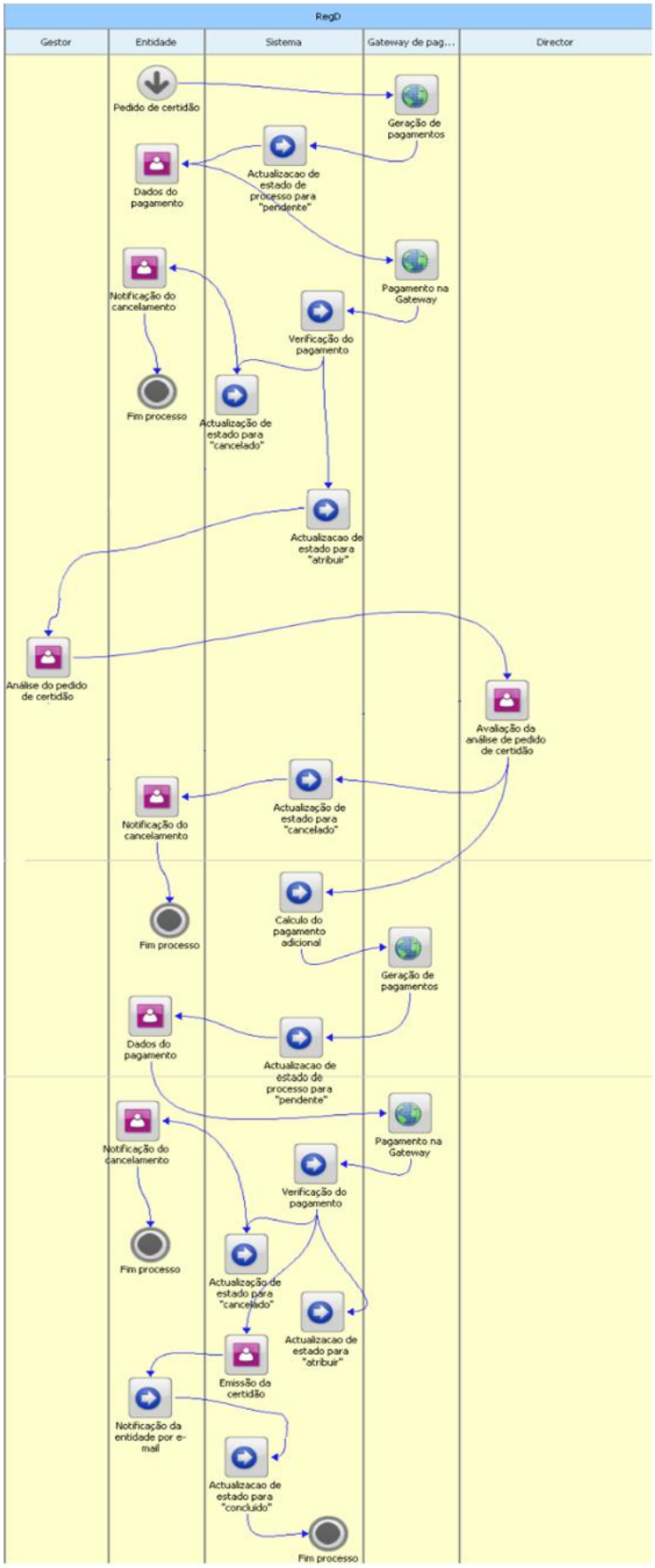


Figura 9 – Pedido de certidão

Todo este processo pode ser novamente despoletado mesmo que o dispositivo em causa já tenha um pedido de certidão. Neste caso, no momento de pesquisa de dispositivos, aparece um *link* "Reemissão".

4.2.3 Requisitos

Após a análise da descrição efectuada, apresenta-se um esquema onde são definidos os passos do ciclo de vida do processo de certidão, assim como todos os caminhos alternativos, actores, pré-condições e pós-condições.

Pré-condições
1. As credenciais de acesso da entidade têm de estar correctas.
2. A entidade tem de ter definido, nas permissões da *role*, o acesso à área de pedido de certidão.
3. Só serão considerados, na pesquisa de dispositivos, os dispositivos que foram submetidos.

Pós-condições
1. O pedido de certidão é submetido e validado por parte do gestor.
2. O gestor analisa o pedido e emite as certidões para os dispositivos assinalados.

Caminho Principal
1. A entidade acede à opção de pedido de certidão.
2. Pesquisa pelos seus dispositivos, selecciona os dispositivos para os quais pretende pedir uma certidão.
3. Submete o pedido de certidão.
4. O sistema valida que os dispositivos

5. O sistema gera os dados de pagamento associados a um pedido de certidão.
6. É criada uma nota de pagamento na *gateway* de pagamentos.
7. Os dados de pagamento são enviados à entidade para que esta efectue o pagamento (referência multibanco, cartão de crédito, transferência bancária ou valores).
8. A entidade efectua o pagamento do pedido de certidão.
9. É gerada uma tarefa para BO para análise do pedido de certidão, para o gestor.
10. O gestor analisa o pedido
11. É gerada uma tarefa para BO para avaliar a análise feita pelo gestor.
12. O director valida a análise feita pelo gestor e aprova.
13. O sistema verifica se existe necessidade de pagamento adicional.
14. Caso não exista, são emitidas as certidões
15. É gerada uma tarefa de emissão de certidão.
16. O gestor recebe uma tarefa de emissão onde terá acesso às certidões
17. O gestor assina e carimba as certidões e o ofício, faz *upload* dos documentos e completa a tarefa.
18. A entidade é notificada, via e-mail, de que as certidões já foram emitidas e que estão disponíveis para serem levantadas.
19. O processo é concluído.

Caminho Alternativo – 1

1. No passo 6 do caminho principal, a entidade não efectua o pagamento inicial do pedido de certidão no prazo de um mês.
2. O processo passa para o estado de "Cancelado".
3. É enviado para a *gateway* de pagamentos uma nota de cancelamento sobre o pagamento existente.
4. A entidade é notificada que não efectuou o pagamento.

5. O processo passa para o estado de “Cancelado”.
6. O processo termina.

Caminho Alternativo – 2

1. O gestor, no passo 11 do caminho principal, recusa o pedido de certidão.
2. Ao submeter é criada uma tarefa em BO.
3. O processo vai ao director para validação.
4. A entidade é notificada do motivo da recusa por parte do gestor.
5. O processo de pedido de certidão muda de estado para “Concluído”.
6. Não há emissão de certidão.

Caminho Alternativo – 3

1. No passo 13 o sistema verifica que existe pagamento adicional.
2. É criada uma nova nota de pagamentos na *gateway* de pagamentos.
3. São geradas novas referências multibanco e enviadas à entidade.
4. O sistema verifica se foi efectuado ou não o pagamento adicional.
5. O pagamento adicional não foi efectuado pela Entidade num período de um mês.
6. O processo de pedido de certidão muda de estado para “Concluído”.
7. Não há emissão de certidão.

De seguida é apresentado um diagrama ilustrativo do caminho principal do processo, Figura 10.

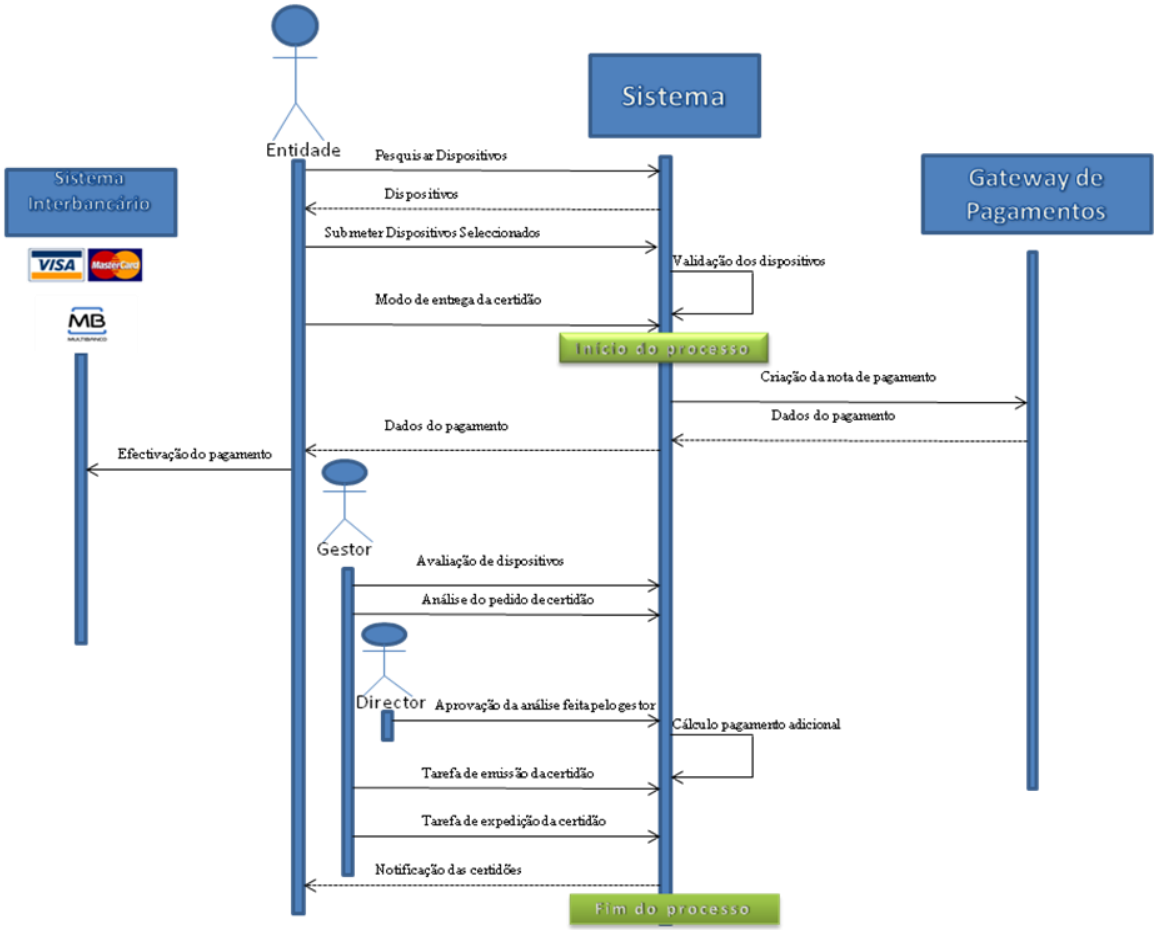


Figura 10 – Diagrama do pedido de certidão

4.2.4 Modelo de dados

Neste subcapítulo será apresentado o modelo de dados utilizado no processo de pedido de certidão. Como o pedido de certidão é um dos processos centrais na aplicação, o modelo de dados, apresentado de seguida, é apenas parcial (Figura 11).

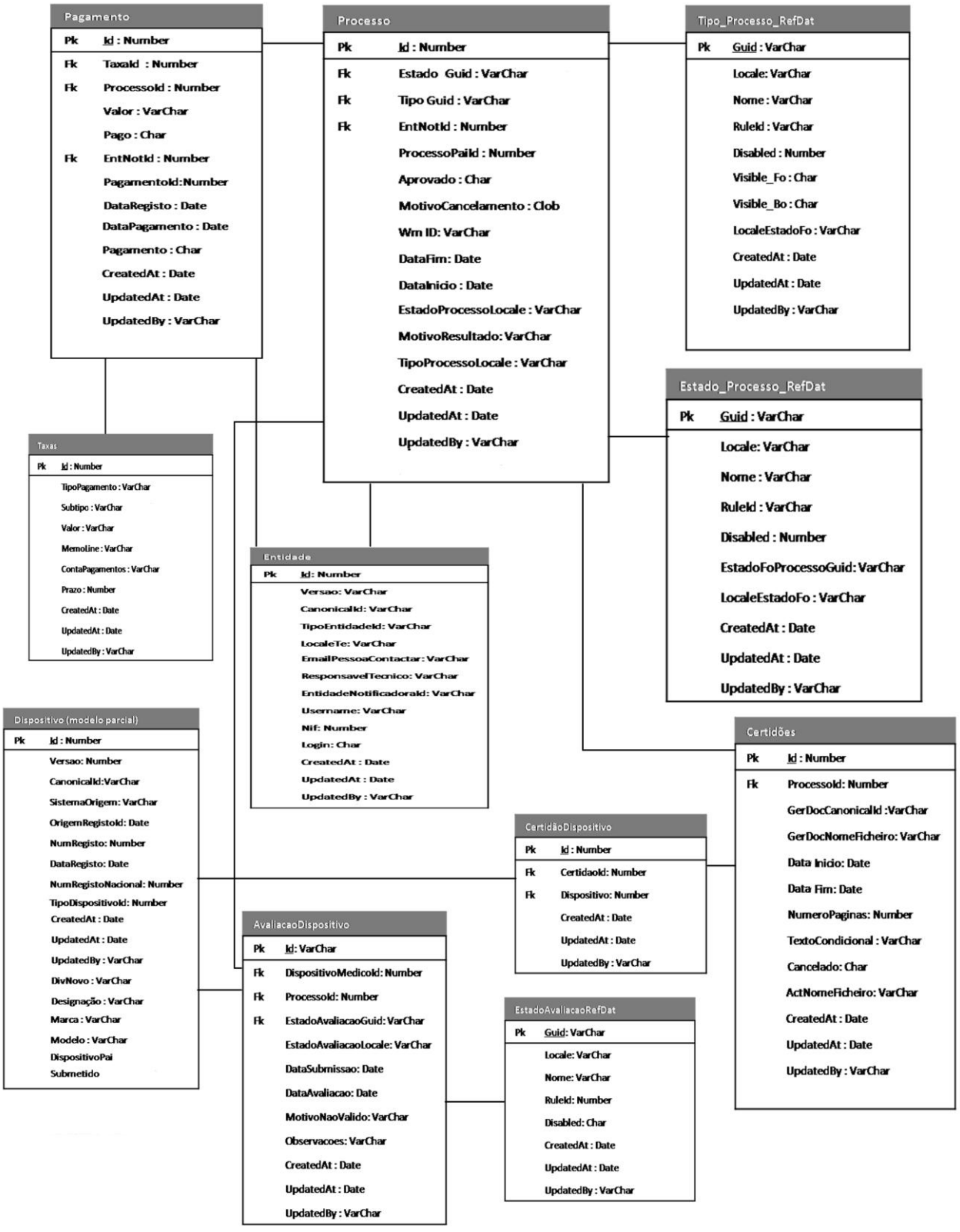


Figura 11 – Modelo relacional

4.2.1 Desenvolvimento do processo BPM

Após análise de todo o desenho do processo, o ciclo de vida do processo é apresentado na Figura 10 e uma pequena descrição dos passos mais importantes na Tabela 12.

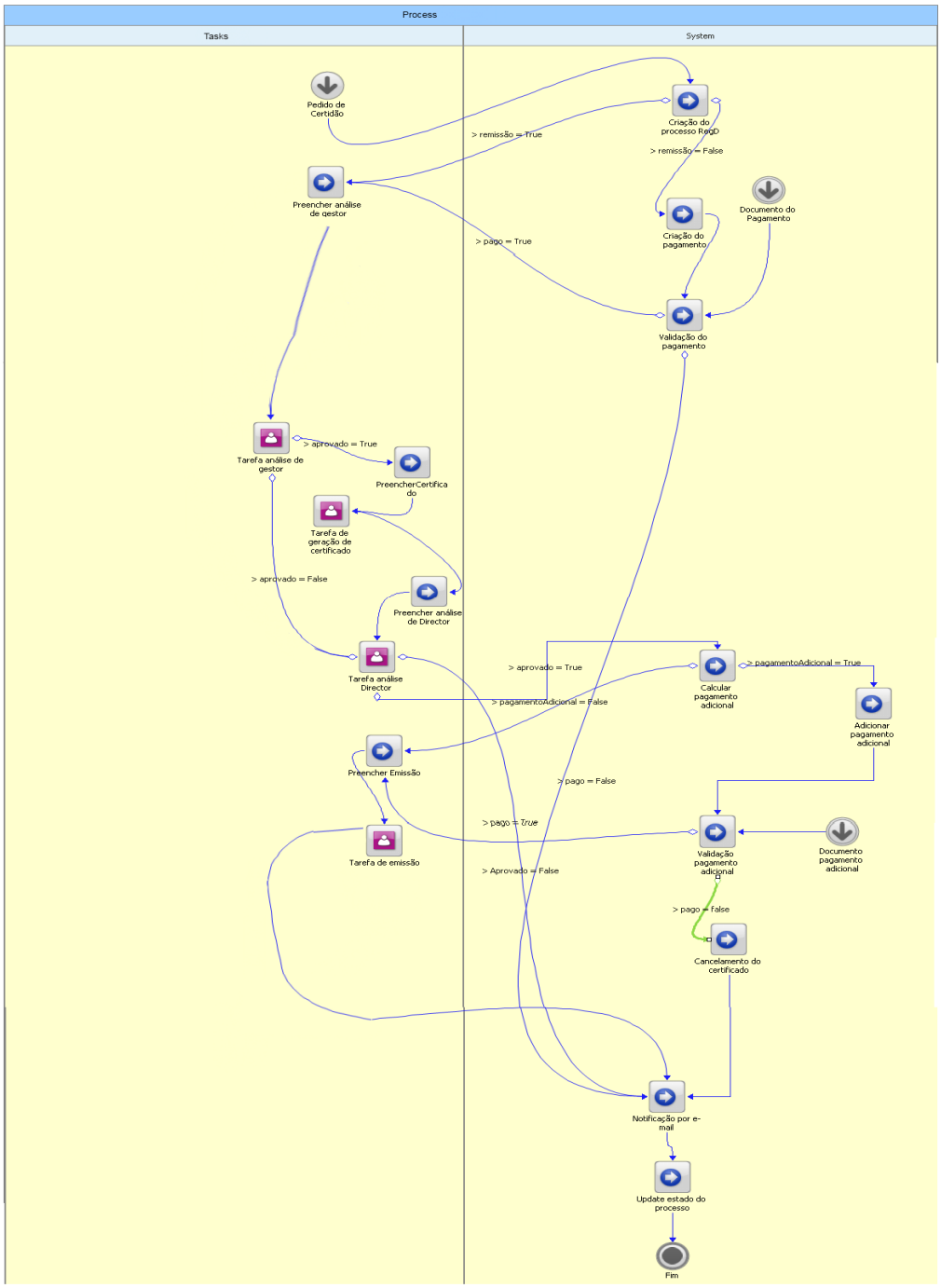


Figura 12 – Processo de pedido de certidão

Etapas importantes	Descrição
Pedido de certidão	O processo de pedido de certidão é despoletado neste passo. Este passo ocorre quando o utilizador faz a submissão do pedido de certidão.
Criação do pagamento	O passo de criação e validação do pagamento tem sincronização com a *gateway* de pagamentos.
Validação do pagamento	O passo da validação do pagamento possui uma *Scheduler Task* que de 5 em 5 minutos verifica na *gateway* de pagamentos se o pagamento foi efectivado.
Documento do pagamento	Este documento é enviado pela *gateway* de pagamentos quando o pagamento é realizado.
Fim	Este é o último passo do processo de pedido de certidão. Com este passo o processo passa para o estado de "Concluído" independentemente de ter sido com sucesso ou não.

Tabela 2 – Passos importante do pedido de certidão

4.2.2 Desenvolvimento e implementação dos *Web-Services*

Antes de desenvolver os serviços que foram utilizados no processo de pedido certidão, criaram-se os documentos que se iriam utilizar, Figuras 13, 14 e 15. Estes documentos são objectos que facilitam a transição de dados entre vários serviços e ajudam na implementação das tarefas que utilizam os serviços e documentos criados.

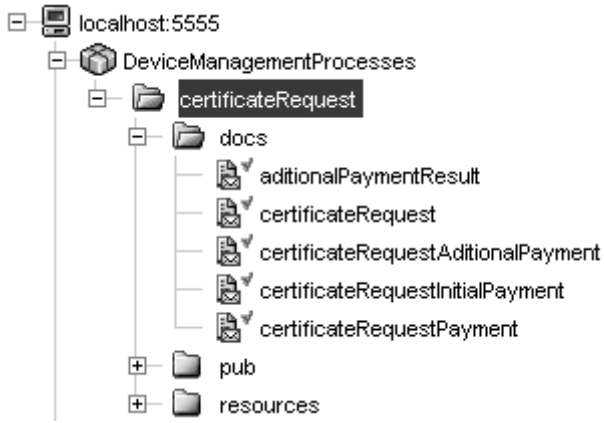


Figura 13 – Estrutura dos documentos usados no pedido de certidão

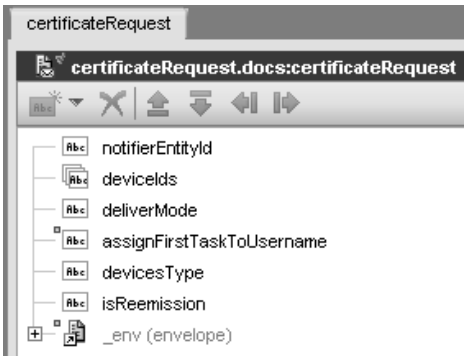


Figura 14 – Documento de pedido de certidão

Figura 15 – Documento de pagamento

Todos os documentos criados ou modificados têm que fazer a sincronização com o *Broker* do IS de modo a ficarem disponíveis.

Os *Web-Services* desenvolvidos no âmbito do processo encontram-se na pasta *pub* , Figura 16. Alguns serviços utilizados no processo do pedido de certidão foram desenvolvidos no âmbito de outros processos e reutilizados para este processo.

Figura 16 – Serviços principais

A pasta *payment* (Figura 17) contém serviços de acesso à *gateway* de pagamentos. Qualquer processo pode utilizar estes serviços mas, no caso do RegD, o único processo que contém pagamentos é o pedido de certidão.

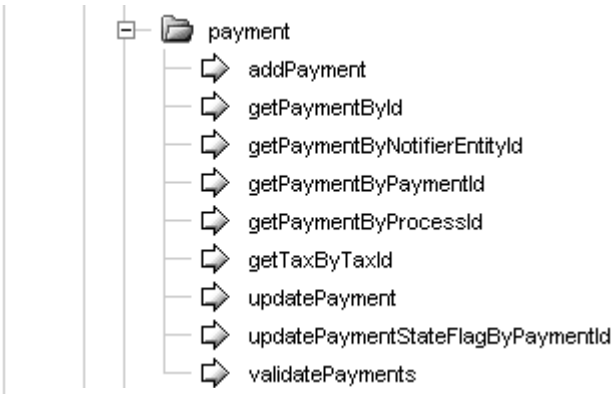


Figura 17 – Serviços de acesso à *gateway*

De seguida serão explicados alguns serviços importantes que foram utilizados no âmbito do processo do pedido de certidão. Alguns dos serviços já estavam desenvolvidos, outros já estavam criados mas tiveram que ser modificados e outros ainda foram criados de raiz.

Conforme já mencionado no relatório, cada serviço tem os seus parâmetros de entrada e os seus parâmetros de saída. Estes parâmetros podem ser *Strings*, valores numéricos ou inclusive documentos (ver Figuras 11, 12, 13). Estes serão respectivamente os valores de entrada na *pipeline* do serviço e os valores de saída da *pipeline* do serviço.

4.2.3 getCertificatesByProcessId

O serviço getCertificatesByProcessId vai buscar todos os certificados dado o *id* de processo. Tem como parâmetros de entrada e de saída os valores apresentados nas Figuras 18 e 19.

Figura 18 – Parâmetros de entrada

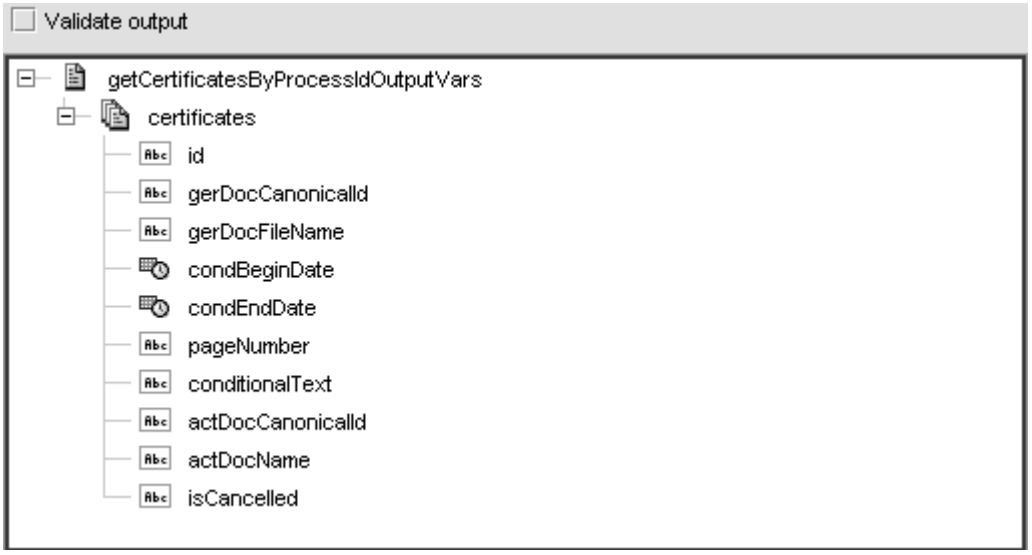


Figura 19 – Parâmetros de saída

Na figura seguinte (Figura 20) encontra-se a estrutura do serviço. Aqui estão definidos os passos necessários para o serviço devolver os certificados.

O serviço *getCertificatesByProcessId*, além dos mapeamentos, apenas invoca um *adapter*. Este *adapter*, como já foi descrito, faz uma invocação à base de dados. Na zona assinalada da imagem pode ver-se as transformações (setas) que existem a nível da *pipeline*.

O serviço *notificationByEmail* é o serviço responsável pela notificação das entidades por e-mail. Tem como parâmetros de entrada os valores mostrados na figura 19.

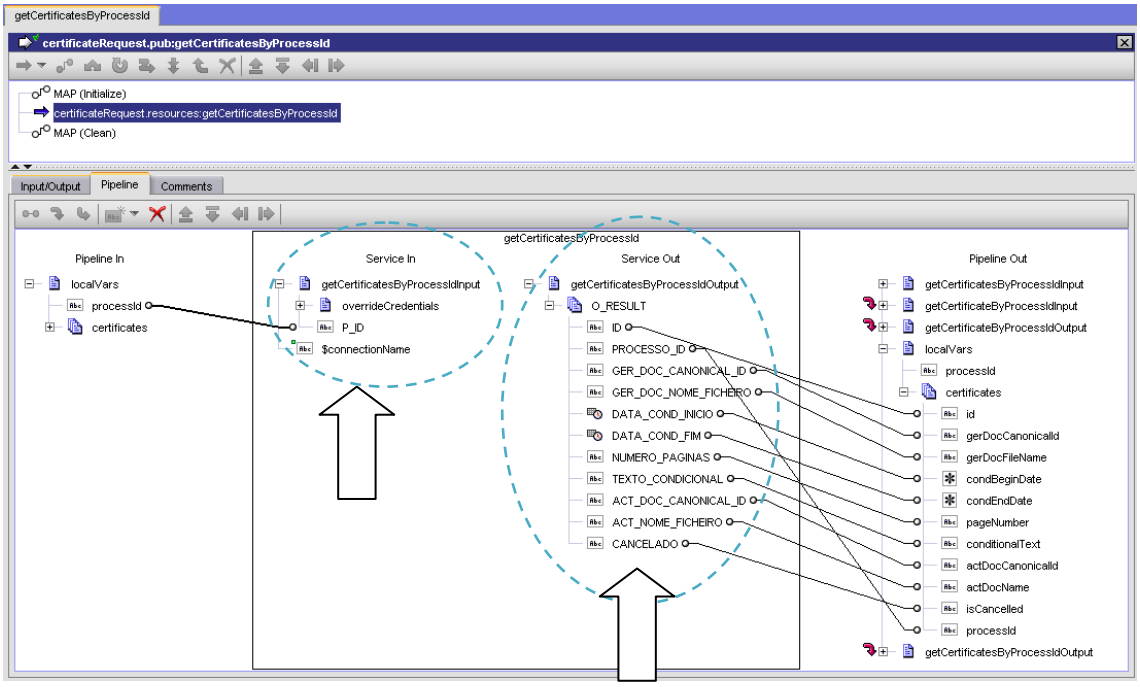


Figura 20 – Estrutura do serviço

O serviço recebe maioritariamente documentos como parâmetros, criando desta forma o e-mail para enviar à entidade, Figura 21.

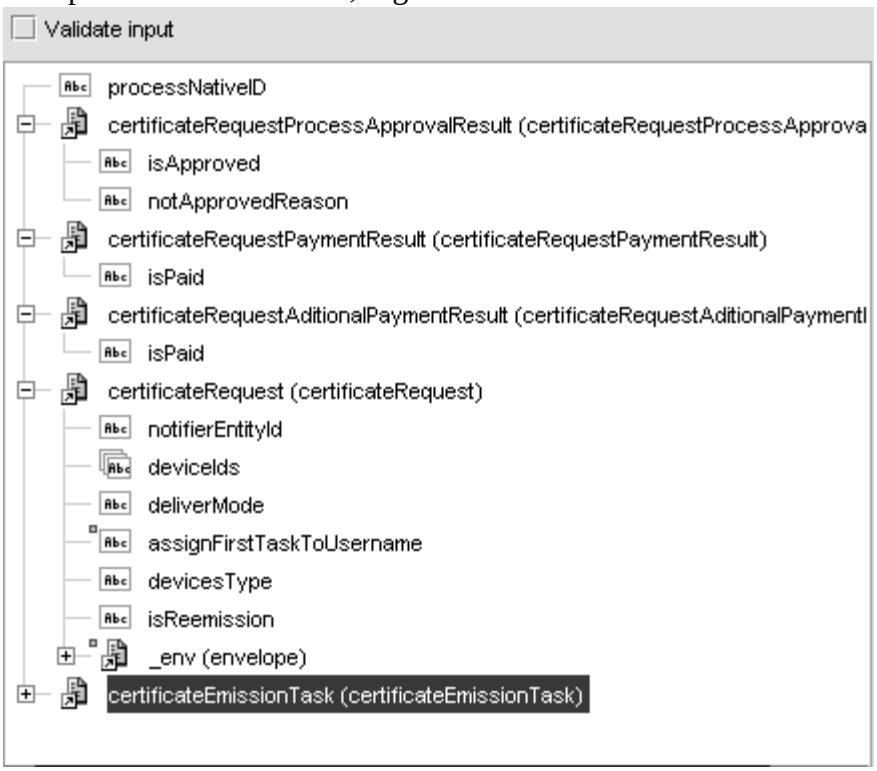


Figura 21 – Parâmetros de entrada

De seguida é apresentada a estrutura do serviço *notificationByEmail*, Figura 22.

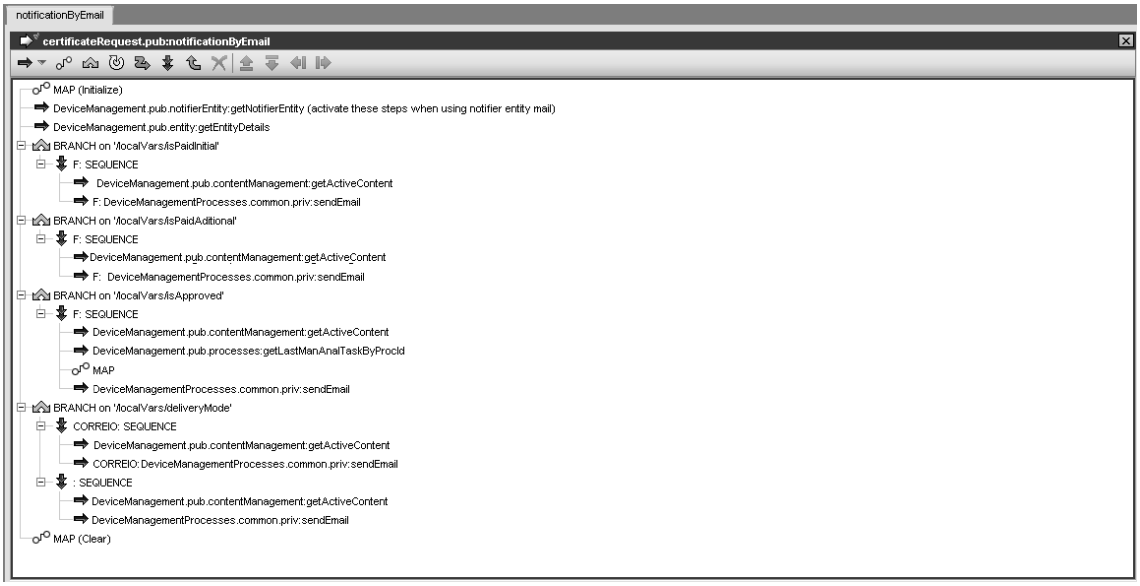


Figura 22 – Estrutura do serviço notificationEmail

Este serviço invoca o serviço *sendEmail*, que por sua vez invoca um servidor de e-mail que envia e-mails por SMTP. Durante a invocação do serviço são efectuadas várias verificações (no passo dos *Branch*) descobrindo desta forma que tipo de e-mail terá que ser enviado à entidade.

Na Figura 23 vemos um exemplo da *pipeline* quando se invoca o serviço *sendEmail*. Os parâmetros de entrada do serviço *sendEmail* são em tudo semelhantes aos dados a preencher quando queremos enviar um e-mail comum.

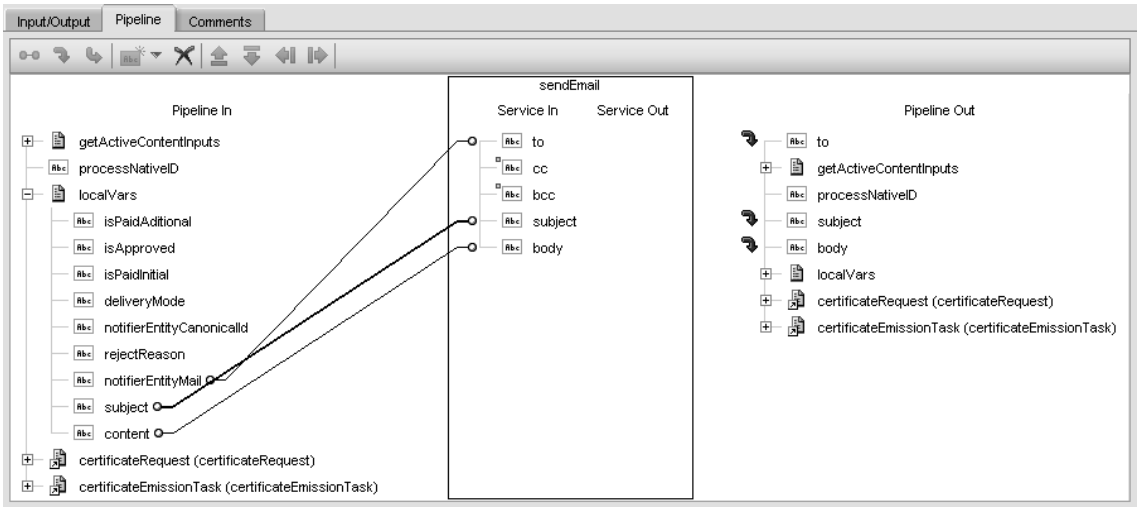


Figura 23 – Pipeline sendEmail

4.2.4 getDeviceInfoList

O serviço *getDeviceInfoList* tem como objectivo devolver toda a informação de um determinado dispositivo de um processo. Este serviço é útil quando queremos obter todos os dispositivos que estão no âmbito do processo de pedido de certidão.

Este serviço tem como parâmetros de entrada e como parâmetros de saída os valores que estão da Figura 24 e 25.

Figura 24 – Parâmetros de entrada

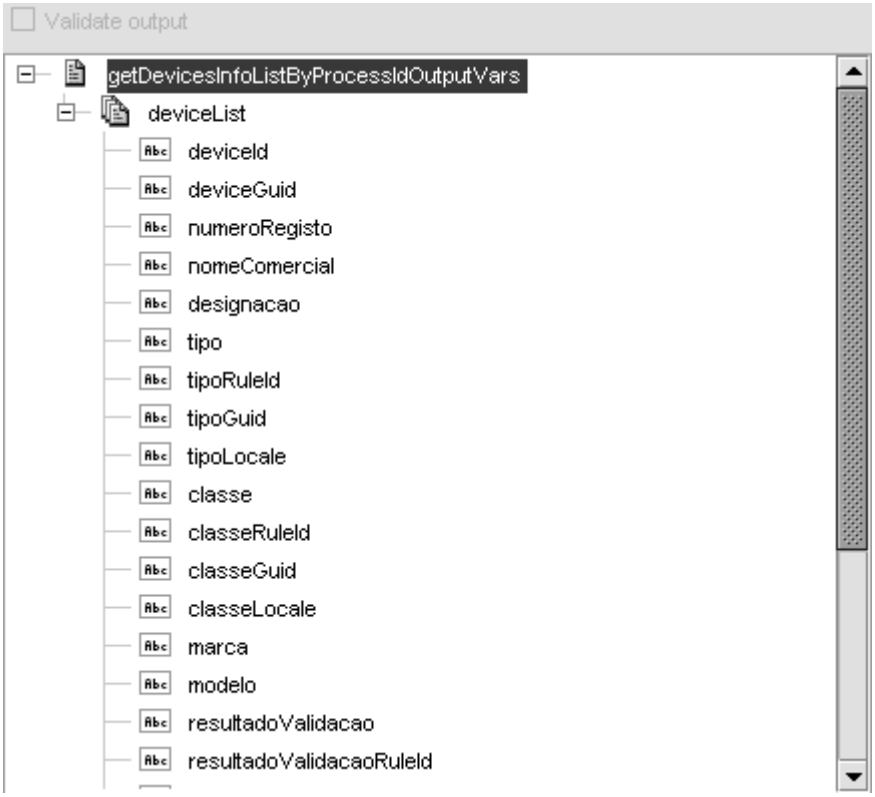


Figura 25 – Parâmetros de saída

Como parâmetro de entrada o serviço apenas precisa do *id* do processo. Como parâmetro de saída devolve uma lista de documentos com a informação do dispositivo.

A Figura 26 mostra a estrutura do serviço getDevicesListByProcessId.

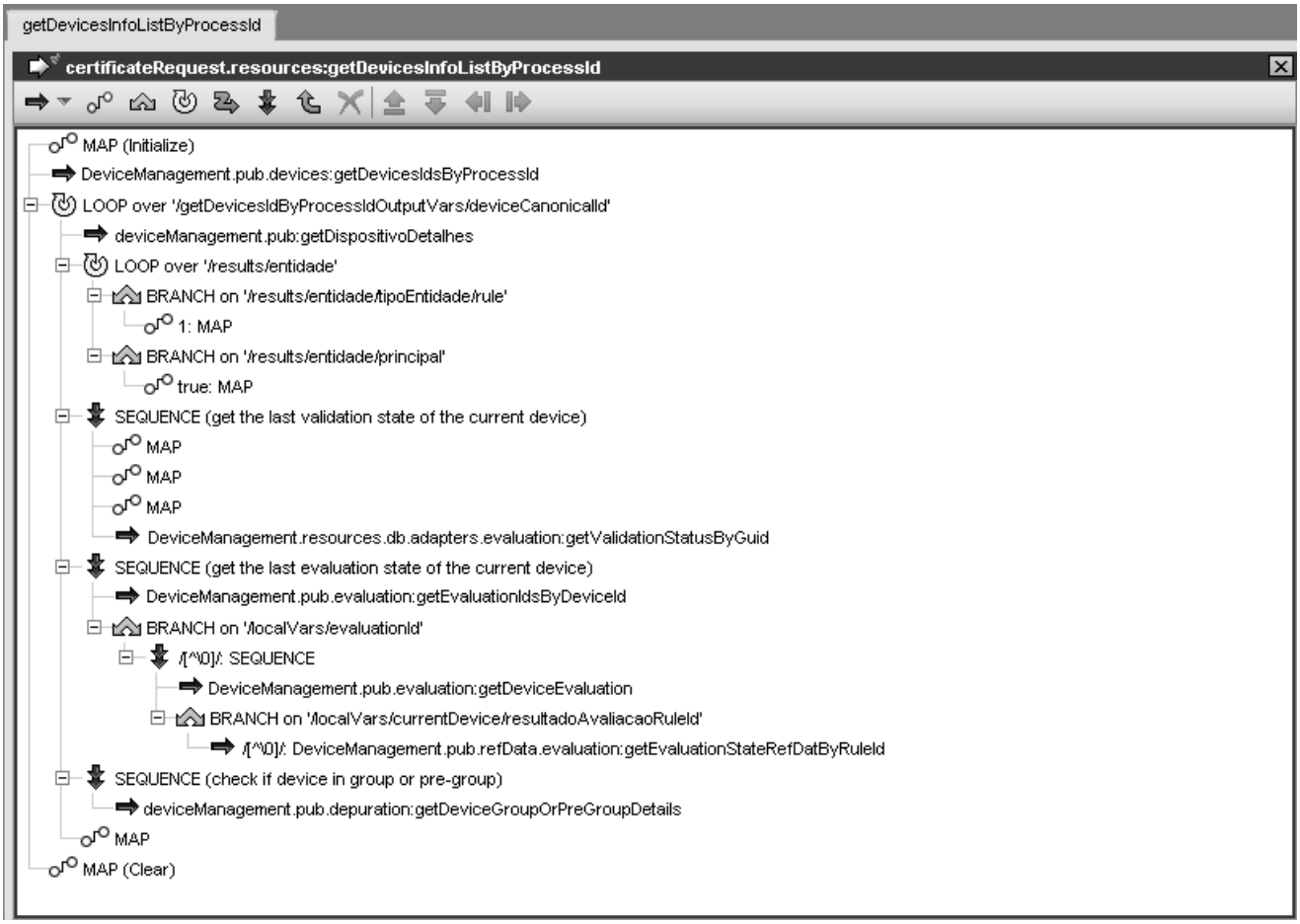


Figura 26 – Estrutura do serviço getDevicesInfoListByProcessId

Este serviço é complexo, pois a informação sobre o dispositivo encontra-se dispersa e para a obter é necessário utilizar vários recursos. Por exemplo, saber que tipo de entidade está associada ao dispositivo e saber qual foi a última avaliação do dispositivo são informações que não se encontram no mesmo recurso.

4.2.5 ValidatePayment

O serviço *ValidatePayment* é um dos mais complexos do pedido de certidão. Invoca serviços da *gateway* de pagamentos e faz inúmeras verificações de dados. Faz *pooling* para saber se os pagamentos foram concretizados na *gateway* de pagamentos. Para automatizar esta verificação foi desenvolvida uma *ScheduledTask* de modo a invocar este serviço num determinado intervalo de tempo.

Este serviço não tem qualquer parâmetro de entrada nem parâmetro de saída, apesar de publicar um documento do pagamento. Este documento está descrito no ciclo de vida do processo como o passo “Documento do pagamento”.

De seguida é apresentada toda a estrutura do serviço *validatePayments* , Figura 27.

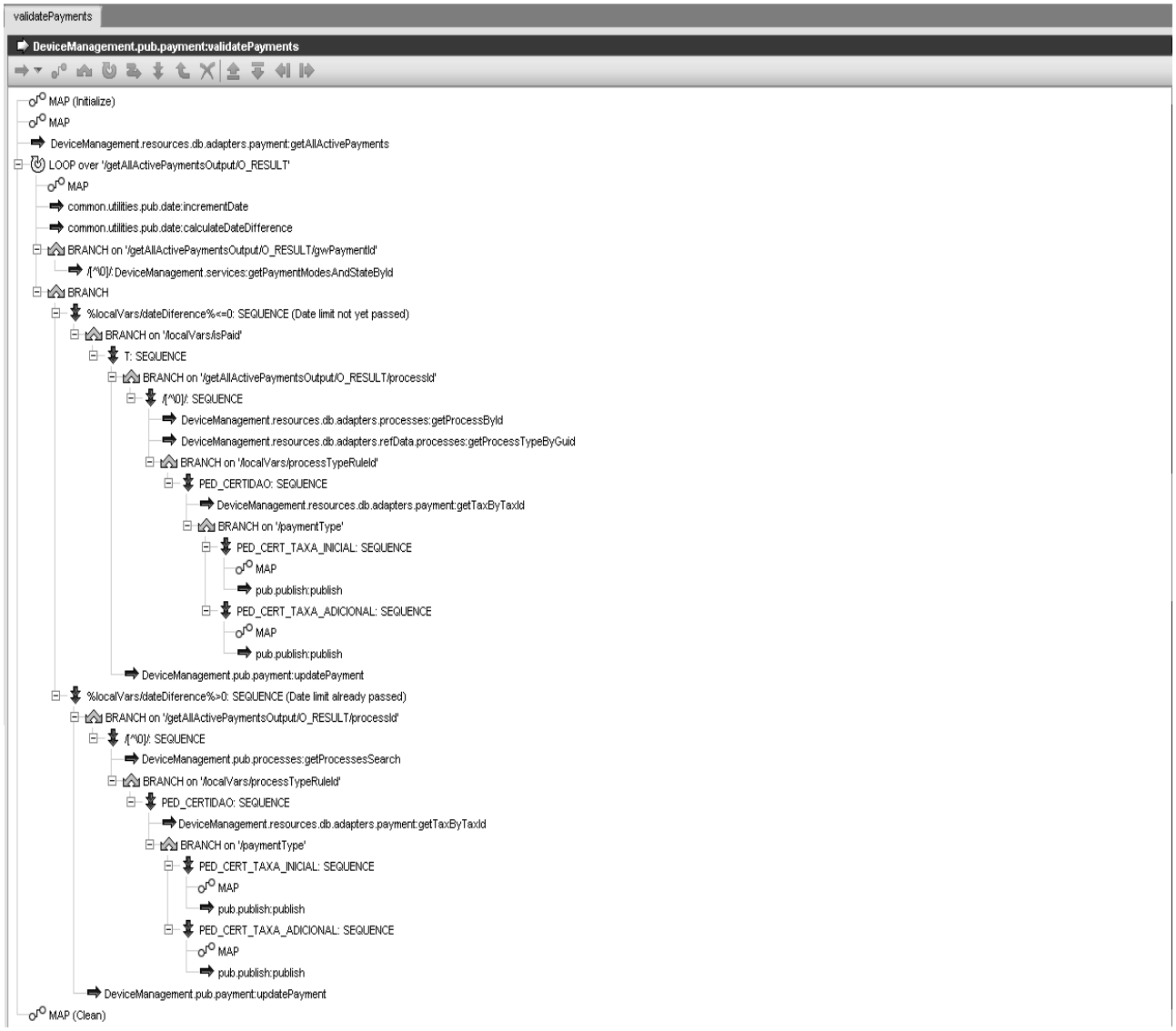


Figura 27 - Estrutura do serviço validatePayment

Este serviço faz, tanto a verificação do pagamento inicial, como o pagamento adicional. Efectua igualmente a actualização de estados do pagamento tanto do lado do RegD como da *gateway* de pagamentos. A publicação do documento de pagamento está no passo "*pub.publish:publish".* Neste passo é enviado um documento para o *Broker* e, deste modo, dá-se a continuação do processo de pedido de certidão.

4.2.6 Desenvolvimento e implementação das tarefas

Foram desenvolvidas algumas tarefas no pedido de certidão. As tarefas desenvolvidas foram maioritariamente para o gestor, que está encarregue de efectuar a avaliação e análise dos dispositivos e da certidão.

Apenas serão mencionados dois dos ecrãs para uma noção geral do conteúdo desenvolvido.

4.2.7 Tarefa de análise de director

Na tarefa de análise de director é efectuada a análise do pedido de certidão. Esta tarefa é constituída pelas seguintes *views*:

- ✓ *Default.view*;
- ✓ *DeviceList.view*;
- ✓ *InformationRequest.view*;
- ✓ *Validation.view*.

De seguida é apresentado o desenvolvimento da tarefa (utilizando o *Designer*) e o resultado da tarefa executada no *browser*. Este é apenas um excerto da Default.View, Figura 28.

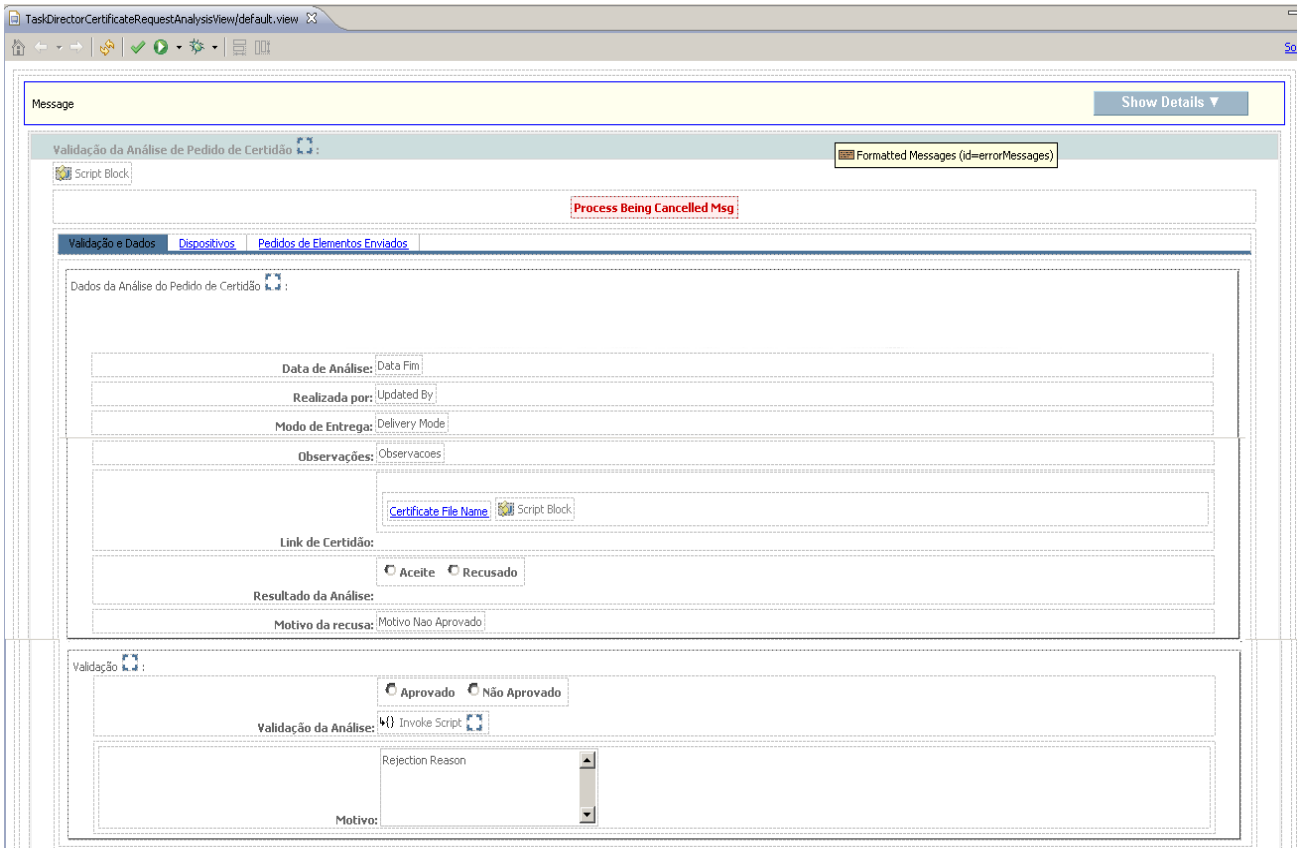


Figura 28 – Default.view

A tarefa contém informação variada sobre o dispositivo e sobre a certidão. Esta informação é disponibilizada através do consumo dos serviços anteriormente desenvolvidos. Na Figura 29 podemos ver o ecrã gerado a partir da Default.view anteriormente criada.no *Designer*.

Figura 29 – Ecrã desenvolvido

4.2.8 FoCreditCardPayment

A tarefa de pagamento por cartão de crédito é complexa dado que comunica com sistemas externos à *RedUnicre*, para validar o cartão de crédito. Também comunica com a *gateway* de pagamentos para efectivar o pagamento. Mais detalhes serão dados no Capítulo 4.4 – G*ateway* de pagamentos.

Esta tarefa apenas tem uma *view*, a Default.view (Figura 30), contudo a *view* possui várias fases. Inicialmente ocorre a fase de inserção dos dados do cartão de crédito onde se pode também consultar os dados do pagamento. Após carregar em "Efectuar o pagamento" passamos para o ecrã de confirmação dos dados de pagamento. Após essa confirmação dos dados pelo lado da Entidade, ao carregar em "Confirmar pagamento" o sistema realiza primeiro a verificação dos dados com a *RedUnicre* e, caso estejam

correctos, efectiva o pagamento (através da *RedUnicre*) e envia essa informação para a *gateway* de pagamentos.

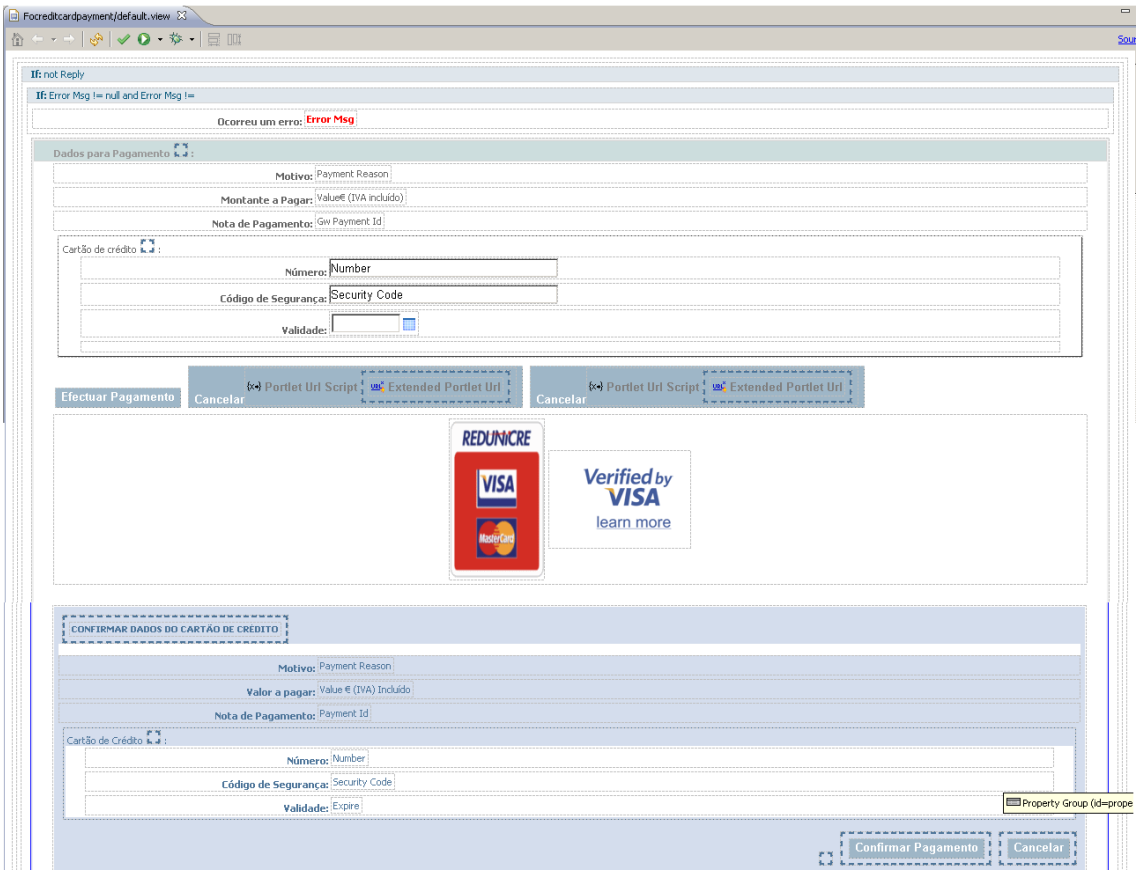


Figura 30 – Desenvolvimento do ecrã de pagamento

Como podemos observar na Figura 31, existem componentes que não são mostrados no ecrã principal (mas que podemos ver no desenvolvimento do ecrã). Isto acontece porque esses componentes só irão aparecer após ocorrer alguma acção por parte da entidade.

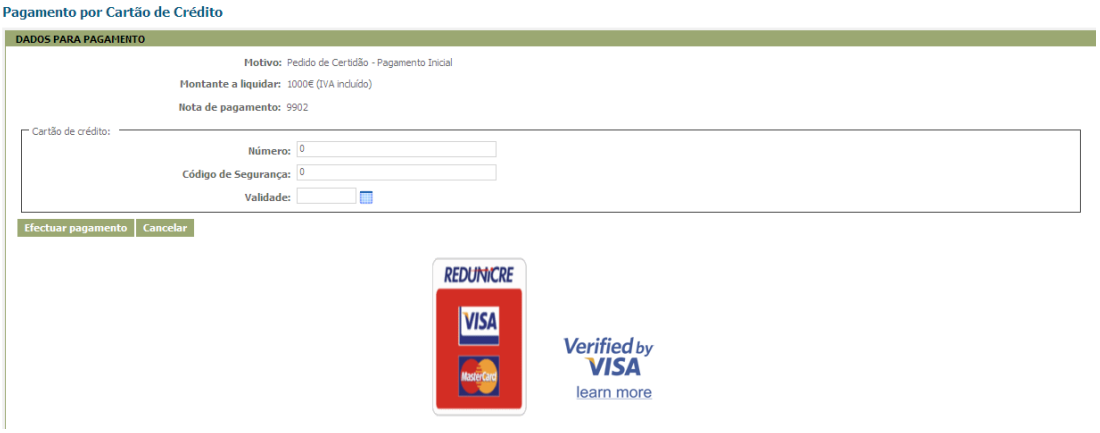


Figura 31 – Ecrã desenvolvido

4.2.9 *Scheduler* de pagamento

Existem *Scheduled Tasks* necessárias para a aplicação funcionar correctamente. Para este processo existe uma *Scheduled Task* específica, de modo a verificar se os pagamentos registados na *gateway* foram pagos e se estão ainda dentro do prazo.

Para efectuar a criação e/ou modificação de tarefas é necessário aceder à página de administração do IS, criar uma nova *Scheduled Task* e definir os parâmetros. De seguida mostra-se o *template* de criação de uma *Scheduled Task* (Figura 32) e um exemplo do conteúdo para a *Scheduled Task* (Tabela 3) de verificação de pagamento.

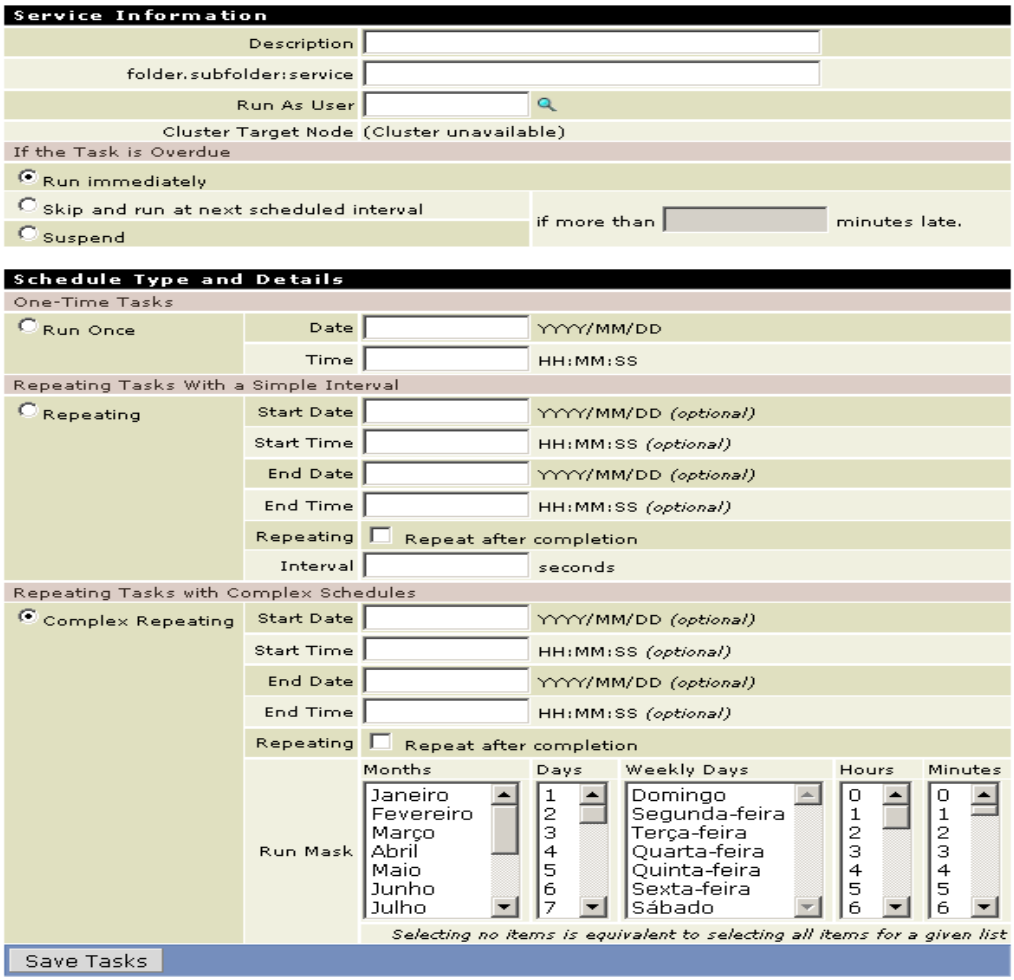


Figura 32 – *Scheduled Task*

Chave da configuração	Valor
Description	Serviço que automaticamente verifica se os pagamentos na *gateway* foram efectuados.
folder.subfolder:service	DeviceManagement.pub.payment:validatePayments – Serviço que é invocado, neste caso o serviço de validação de pagamentos.
Run As User	Administrator.
If the Task is Overdue – Se a *Scheduled Task* chegar ao fim e não obter resposta, volta a ser executada novamente.	

Run immediately	
Repeating Tasks With a Simple Interval – Intervalo de tempo que a *Scheduled Task* se repete.	
Start Date	2011/02/16
Start Time	11:04:48
Repeating	*Repeat after completion*.
Interval	60 – Intervalo de tempo em segundos

Tabela 3 – Exemplo de configuração

4.3 Implementação do processo de registo de utilizador (*Milestone* 1)

4.3.1 Âmbito e especificação do processo

Este processo pretende descrever as acções necessárias para que um utilizador consiga registar-se no portal da aplicação, podendo posteriormente ter acesso de entidade e assim conseguir registar os seus dispositivos e fazer, além de outras coisas, um pedido de certidão.

Este processo, como o pedido de certidão, é desenvolvido em *flow*. Segue-se uma descrição do ciclo de vida do processo de modo a compreender os seus objectivos.

4.3.2 Desenho e descrição

O utilizador acede ao portal do RegD e é-lhe apresentado um primeiro ecrã onde tem a hipótese de efectuar *login*, caso já seja uma entidade registada. Caso de trate de um novo utilizador tem a possibilidade de fazer um novo registo:

- ✓ Ao seleccionar a acção de "Efectuar Registo", surge um ecrã para preenchimento dos dados de registo, nomeadamente nome, morada, e-mail e contactos telefónicos, entre outros que serão mais a frente especificados;
- ✓ Após o preenchimento dos dados é feita a sua submissão. O sistema irá validar a informação submetida e são feitas várias verificações:
    - o É necessário validar se existe a entidade na GestEnt fazendo comparação com NIF. Em simultâneo é executado um algoritmo de verificação de semelhanças entre a entidade que

se pretende registar e todas as entidades registadas na GestEnt. No caso de existirem entidades com grau de semelhança, é criada uma tarefa em BO para o gestor de modo a poder validar o registo de utilizador. Assim o gestor irá verificar se alguma das entidades com grau de semelhança é realmente igual ao utilizador que se pretende registar no RegD;

- ✓ Caso o gestor associar o utilizador a uma entidade, essa entidade será informada que o registo foi feito com sucesso e recebe também as novas credenciais de acesso. Essas credenciais de acesso são atribuídas e guardadas pela *Apache DS*. Caso a entidade não corresponda a nenhuma entidade que resultou da pesquisa, o gestor poderá criar uma nova entidade na *GestEnt* ou cancelar o registo;
- ✓ Ao ser criada uma nova entidade as credenciais de acesso são atribuídas e guardadas pela *Apache DS* e a entidade é notificada por e-mail com as suas credenciais de acesso;
- ✓ Caso o gestor cancele o registo de entidade, o utilizador será informado de que o seu registo não é válido;
- ✓ A entidade tem ao seu dispor uma área onde pode efectuar a alteração dos seus dados pessoais, sendo que alguns dados não poderão ser alterados com o nome e NIF.

4.3.3 Requisitos

Após a análise da descrição efectuada, elaborou-se um esquema onde são definidos os passos do ciclo de vida do processo de registo de entidade, assim como todos os caminhos alternativos, actores, pré-condições e pós-condições.

Actores: Utilizador, Entidade e Sistema.

Pré-Condições
1. A entidade tem de ter acesso à página de *login* do RegD.
2. A entidade tem de ter acesso à acção de "Efectuar Registo" no portal RegD.

Pós-Condições
1. O registo da entidade é criado e esta recebe, no seu e-mail, as credenciais de acesso ao portal: *username* e palavra-passe. 2. Nos casos de já existir, será criada uma tarefa para o gestor poder comparar as semelhanças entre as entidades e decidir associa as entidades existentes na GestEnt, se cancela ou se cria uma nova entidade.

Caminho Principal
1. A entidade acede ao portal RegD. 2. Selecciona a acção de "Efectuar Registo". 3. Insere os dados de registo. 4. Submete o registo e é iniciado o processo de registo de entidade. 5. É validado através do NIF e em simultâneo o algoritmo de comparação de que o utilizador não corresponde a nenhuma entidade já existente na GestEnt. 6. É criada a nova entidade na GestEnt. 7. São geradas as credenciais de acesso: nome de utilizador/palavra-passe na Apache DS. 8. É gerado um e-mail com as credenciais de acesso e enviado à entidade.

Caminho Alternativo – 1
1. No ponto 4 do caminho principal não estão preenchidos os campos obrigatórios. Neste caso, o sistema apresenta uma notificação a informar qual/quais os campos em falta.

Caminho Alternativo – 2
1. No ponto 5 é validado se a entidade existe na GestEnt: comparação de nome.

2. A entidade existe no GestEnt.
3. Surge a tarefa de "Validação do Registo do Utilizador" em BO para o gestor.
4. O gestor vê as semelhanças entre as entidades e associa a entidade que se registou à entidade que está no GestEnt.
5. O gestor opta por criar uma nova entidade.
6. É criada a nova entidade no GestEnt.
7. São geradas as credenciais de acesso: utilizador/palavra-passe na *Apache DS*.
8. É gerado um e-mail para a entidade com as credenciais de acesso, com a informação de que o seu registo foi validado e que deverá confirmar os dados de registo.

Caminho Alternativo – 3

1. No ponto 5 do caminho alternativo 2 o gestor opta por cancelar o utilizador que se registou.
2. É gerado um e-mail para o utilizador a informar que o processo de registo de utilizador foi cancelado.
3. O processo de registo de utilizador termina por cancelamento.

Caminho Alternativo – 6

4. No ponto 5 é validado se a entidade existe no GestEnt: comparação de nome.
5. A entidade existe no GestEnt.
6. Surge a tarefa de "Validação do Registo do Utilizador" em BO para o gestor.
7. O gestor vê as semelhanças entre as entidades e associa a entidade que se registou à entidade que está no GestEnt.
8. O gestor opta por substituir a entidade existente.
9. Os dados da entidade são modificados no GestEnt.

10. Os dados de acesso no *Apache DS* são modificados.
11. É gerado um e-mail para a entidade com as novas credenciais de acesso ao portal do RegD.

De seguida é apresentado um diagrama ilustrativo do caminho principal do processo, Figura 33.

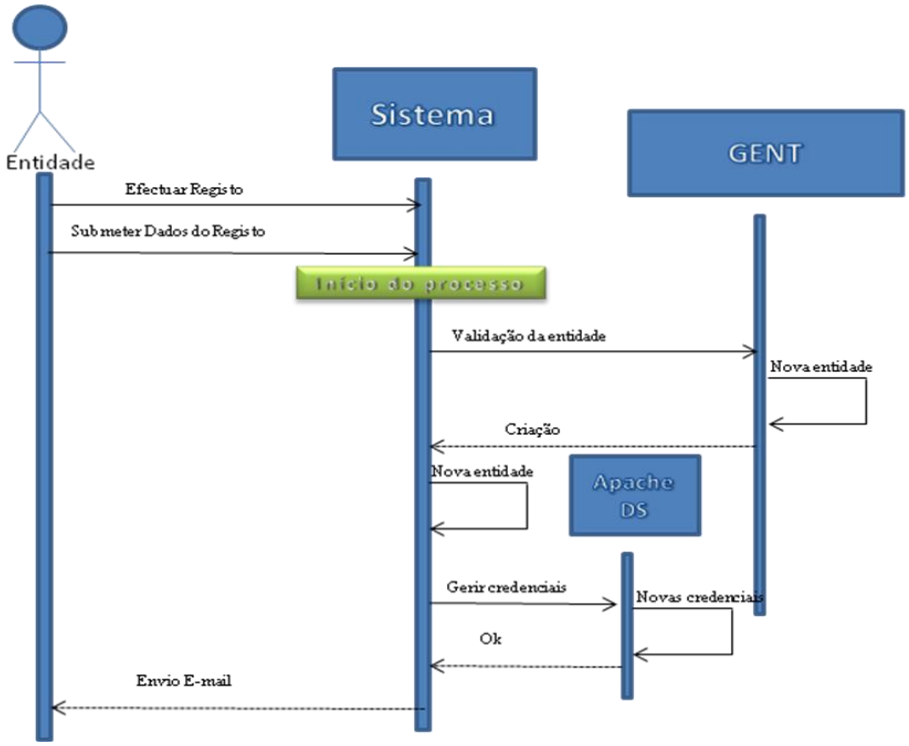


Figura 33 – Diagrama do processo de registo de entidade

4.3.4 Modelo de dados

O modelo de dados aqui apresentado (Figura 34) é o modelo de dados parcial utilizado no processo de registo da entidade. Apesar de ser um processo mais simples do que o pedido de certidão, este processo contém imensa informação, a qual pertence maioritariamente ao utilizador que deseja registar-se.

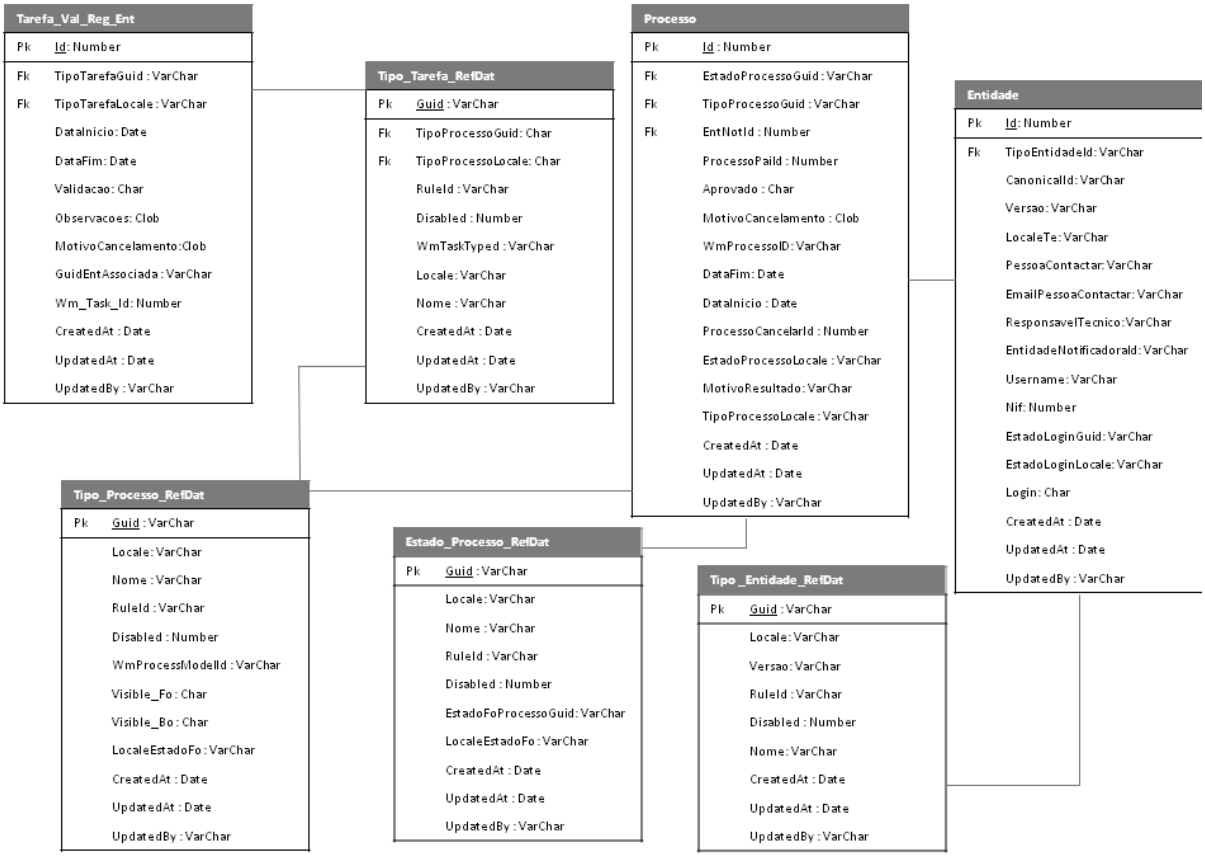


Figura 34 – Modelo de dados do registo de entidade

Após análise de todo o desenho do processo, o ciclo de vida do processo de registo de entidade está descrito na Figura 35. Na Tabela 4 apresenta-se uma pequena descrição dos passos mais importantes do processo.

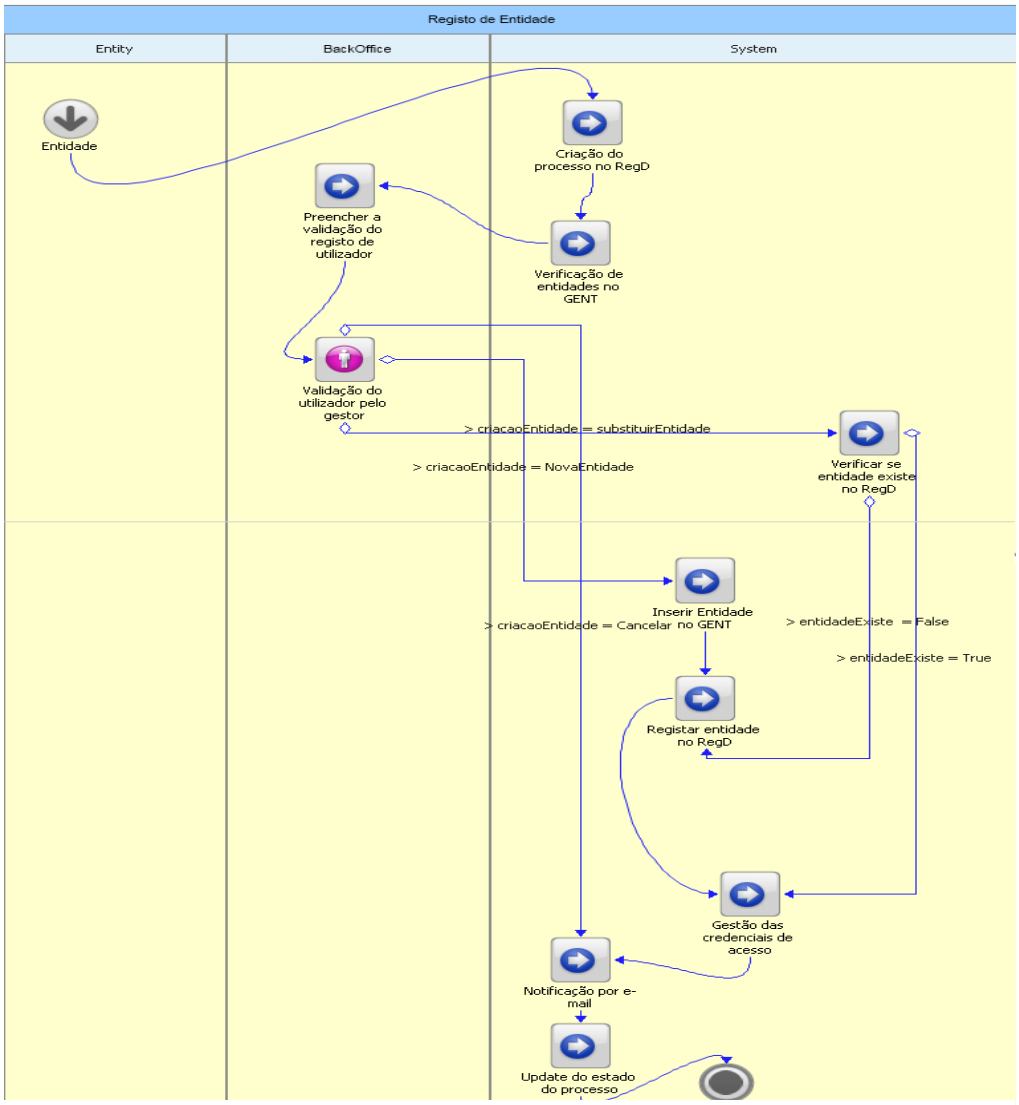


Figura 35 – Processo de registo de entidade

Passos importantes	Descrição
Entidade	O processo é despoletado neste passo. O utilizador preenche os campos de registo de utilizador e ao submeter os dados é iniciado o processo.
Verificação de entidades no GestEnt	Neste passo é efectuada a verificação da existência da entidade no GestEnt. É corrido um algoritmo de semelhanças onde é devolvido o grau de semelhanças entre o utilizador que se quer registar e as entidades já registadas.
Validação do utilizador	Nesta tarefa o gestor analisa o registo de utilizador. Neste passo o gestor pode cancelar o registo de utilizador, criar um novo utilizador ou fazer a

pelo gestor	substituição do utilizador por um já registado (lista devolvida pelo GestEnt).
Gestão das credenciais de acesso	Neste passo é feita uma comunicação com o *Apache Directory* para a gestão das credenciais de acesso do utilizador.

Tabela 4 – Passos importantes do registo de entidade

4.3.5 Desenvolvimento e implementação dos *Web-Services*

O método de desenvolvimento do processo de registo de entidade é em tudo semelhante ao do pedido de certidão.

Inicialmente desenvolveram-se os documentos que dariam suporte aos serviços que seriam criados, Figuras 36, 37 e 38.

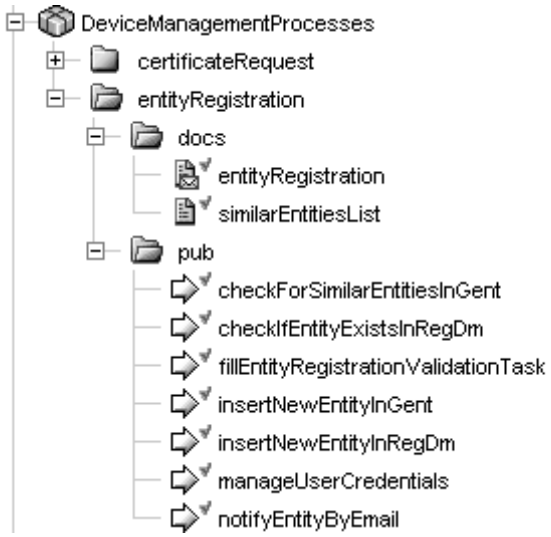


Figura 36 – Documentos do registo de entidade

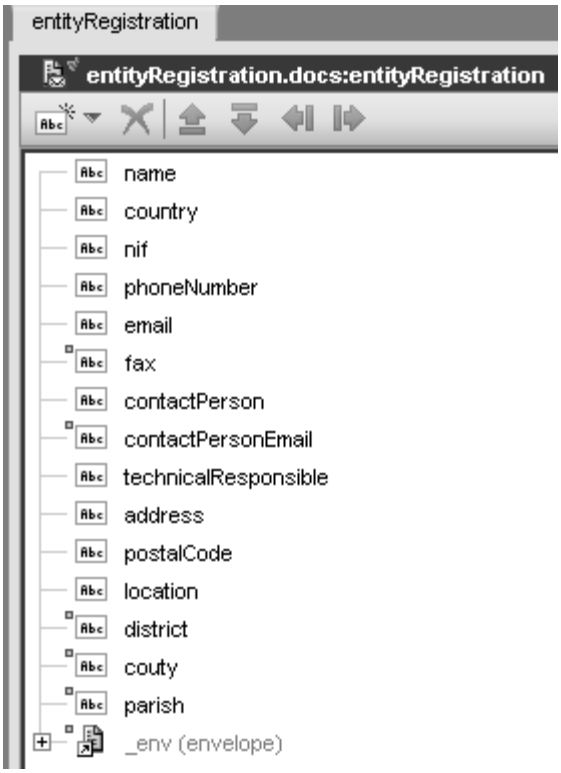


Figura 37 – Documento similarEntitiesList

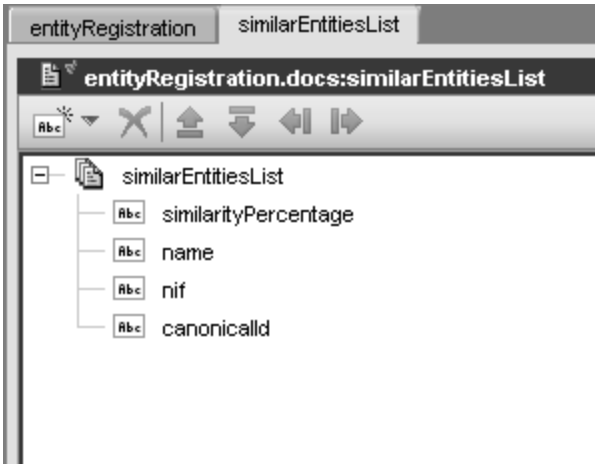


Figura 38 – Documento entityRegistration

Este processo utiliza apenas dois documentos de suporte. Num dos documentos encontra-se guardada toda a informação do utilizador e no outro encontra-se uma lista onde estão definidas as semelhanças entre o utilizador que se quer registar e as entidades já registadas no GestEnt.

A lista de serviços desenvolvida no âmbito do processo encontra-se na pasta *pub*. Nesta pasta estão serviços que fazem a verificação se uma determinada entidade existe no sistema (checkForSimilarEntitiesInGent no caso do GestEnt, e checkIfEntityExistsInRegD no caso do RegD) e que fazem comunicação com o *Apache Directory* (manageUserCredencials).

4.3.6 checkIfEntityExistsInRegD

Este serviço (Figura 39) verifica se uma determinada entidade já existe no sistema do RegD.

Figura 39 – Serviço checkIfEntityExistsInRegD

Este serviço é invocado no passo "Verificar se entidade existe no RegD" do ciclo de vida do processo.

Este serviço tem como parâmetros de entrada e de saída os valores mostrados nas Figuras 40 e 41.

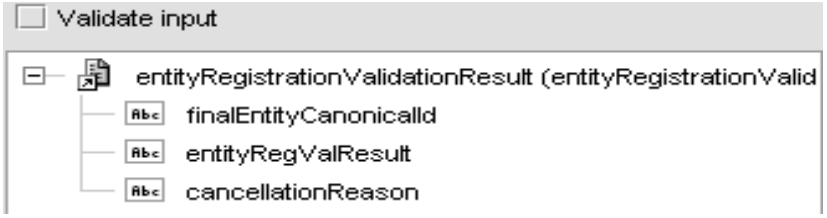


Figura 40 – *Input*

Figura 41 – *Output*

4.3.7 ManageUserCredencials

O objectivo deste serviço (Figura 42) é fazer a comunicação com o sistema de gestão de credenciais, *Apache Directory*.

Figura 42 – Serviço manageUserCredencials

Este serviço serve para criar novas credenciais para um utilizador ou para modificar credenciais já existentes. Os parâmetros de entrada e de saída encontram-se nas Figuras 43 e 44.

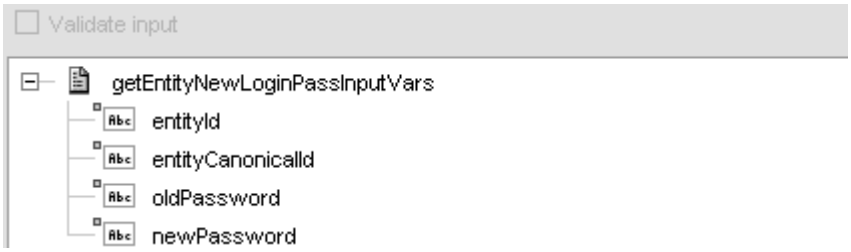


Figura 43 – *Input*

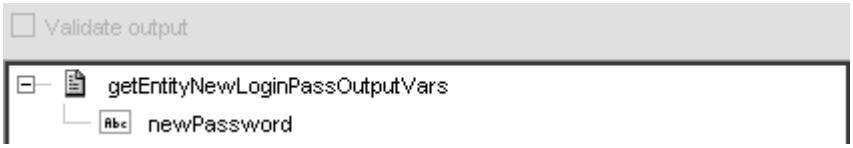


Figura 44 – *Output*

4.3.8 Desenvolvimento e implementação das tarefas

Como se pode verificar no ciclo de vida do processo de registo de utilizador existe apenas uma tarefa, a tarefa de validação do utilizador pelo gestor.

Na tarefa de validação do utilizador pelo gestor, o gestor decide se vai criar uma nova entidade, se vai associar a uma entidade já existente ou se vai cancelar o registo do utilizador. Esta tarefa é constituída pelas seguintes *views*:

- ✓ *Default.view*;
- ✓ *UserInfo.view*;
- ✓ *Validation.view*.

Na Default.view como se pode verificar na Figura 45 apenas faz *import* das outras duas *views*.

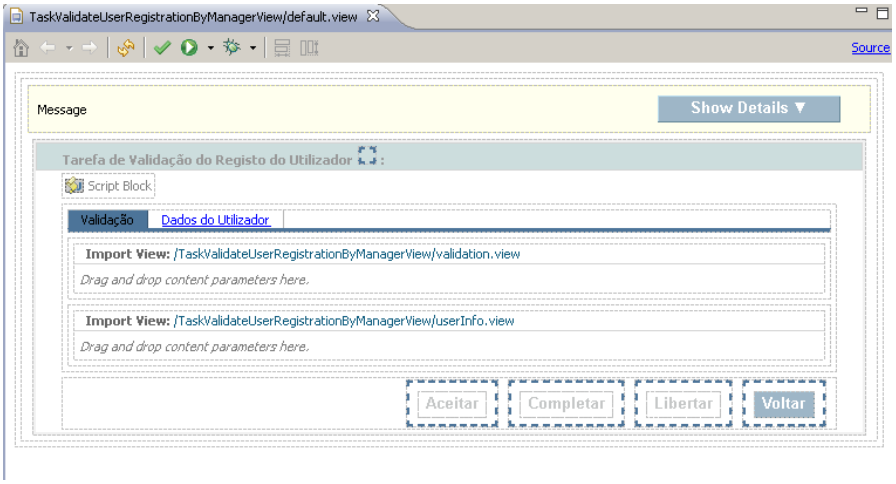


Figura 45 – Default.view

De seguida apresenta-se a *view* Validation.view (Figura 46). É nesta *view* que o gestor decide se cria, associa ou cancela a nova entidade.

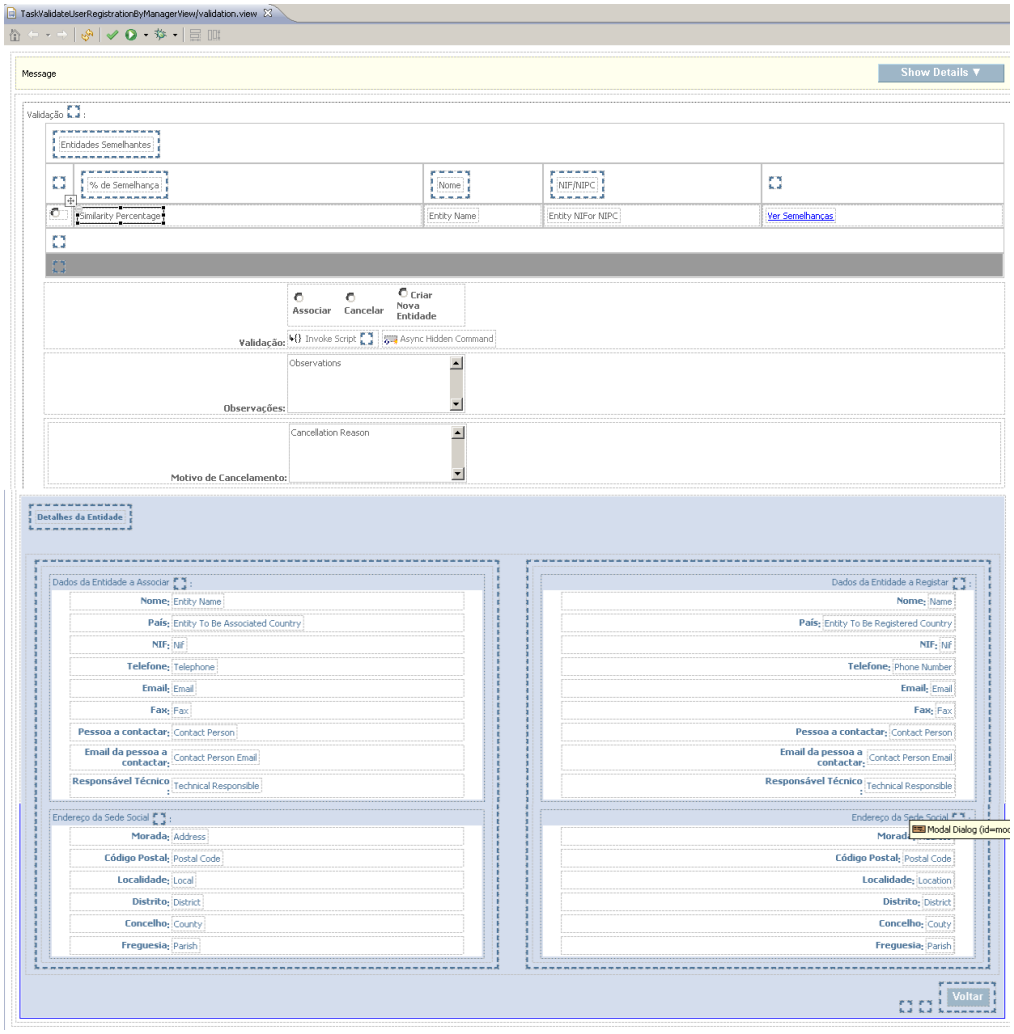


Figura 46 – Validation.view

Aqui podemos ver o ecrã gerado a partir da *view* anteriormente criada, Figura 47.

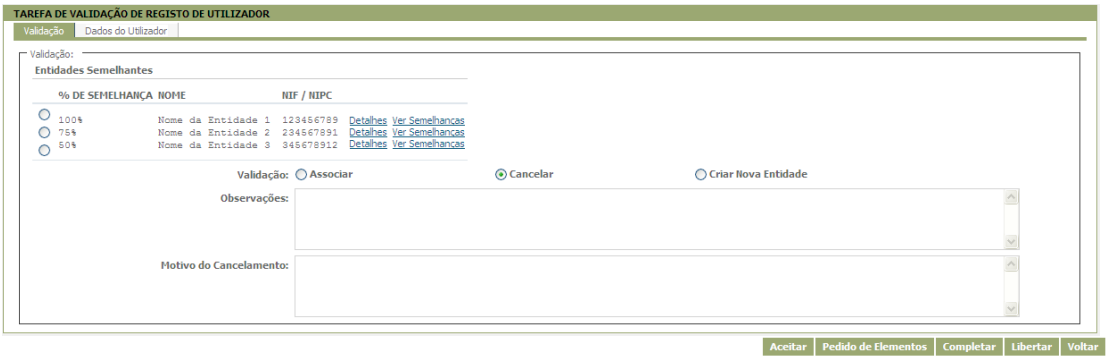


Figura 47 – Ecrã criado

Existe informação na *view* que não está presente no ecrã porque está oculta, por exemplo. Os detalhes da entidade definidos na *view* estão acessíveis através do *link*

detalhes que está no ecrã. Ao carregar neste *link* abrir-se-á uma *pop-up* com os detalhes da entidade.

Além desta tarefa foi também desenvolvido o ecrã que despoleta o processo de registo. Este ecrã está disponível quando o utilizador carrega no botão de novo utilizador, Figura 48.

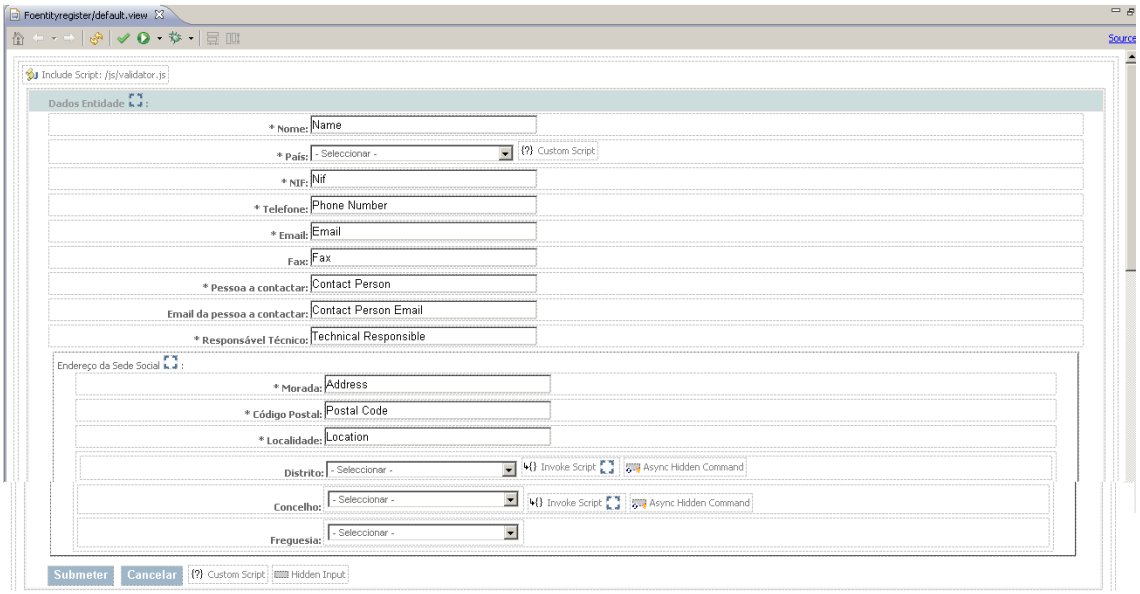


Figura 48 – Default.view

O utilizador preenche todos os dados e carrega no *submeter*. Esta acção despoleta um novo processo de registo de utilizador.

Nos campos distrito, concelho e freguesia encontra-se uma componente de *script.* Estas componentes servem para invocar *scripts* em *JavaScript* de modo a obter a lista de todos os distritos, concelhos e freguesias. Além deste *script* existem *scripts* de validação de conteúdos como por exemplo validação de e-mail, telefone e NIF.

4.3.9 Integração com *Apache Directory*

O processo de registo de utilizador tem uma integração com uma *Apache Directory.* A *Apache Directory* é um servidor de directórios que suporta vários protocolos entre eles, o LDAP. O LDAP é um protocolo utilizado na pesquisa e na actualização de directórios.

A finalidade deste sistema é armazenar e gerir os dados das entidades registadas em qualquer sistema que seja implementado. Neste caso foi necessário recorrer ao *Apache*

Directory para poder gerir as credenciais de acesso dos utilizadores. A interacção com este sistema é através de *WebService connectors*. Estes *WebServices* são usados pelo *Developer* para podermos desenvolver e tirar proveito destes serviços.

De seguida são mostrados alguns comandos utilizados na construção inicial dos directórios.

```
dn: dc=dcName,dc=pt
objectClass: domain
objectClass: top
dc: dcName

dn: ou=groups,dc=dcName,dc=pt
objectClass: organizationalUnit
objectClass: top
ou: groups

dn: ou=users,dc=dcName,dc=pt
objectClass: organizationalUnit
objectClass: top
ou: users

dn: ou=project,ou=groups,dc=dcName,dc=pt
objectClass: organizationalUnit
objectClass: top
ou: project

dn: ou=project,ou=users,dc=dcName,dc=pt
objectClass: organizationalUnit
objectClass: top
ou: project

dn: uid=sysadmin,ou=project,ou=users,dc=dcName,dc=pt
objectClass: organizationalPerson
objectClass: person
objectClass: uidObject
objectClass: inetOrgPerson
objectClass: top
cn: SysAdmin
sn: SysAdmin
uid: sysadmin
userPassword:: sysadmin

dn: cn=externalUsers,ou=project,ou=groups,dc=dcName,dc=pt
objectClass: groupOfUniqueNames
objectClass: top
cn: externalUsers
uniquemember: uid=sysadmin,ou=project,ou=users,dc=dcName,dc=pt
```

Com esta estrutura serão criados os directórios necessários para podermos gerir as credenciais de acesso ao sistema.

Para poder gerir o acesso de uma determinada entidade, ao ser invocado o serviço que gere as credenciais, será necessário passar os seguintes parâmetros:

- ✓ URL (Uniform Resource Locator) do LDAP;
- ✓ Utilizador com credenciais de administrador para poder fazer alterações;
- ✓ Grupo onde queremos inserir/procurar o utilizador;
- ✓ Nome do utilizador;

- ✓ Palavra-passe do utilizador.

No caso de querermos fazer uma pesquisa no directório basta apenas acrescentar o campo *filter* (uid=%localVars/entityCanonicalId% no caso de querermos pesquisar pelo *canonical id*) e ao invocar o serviço ele irá devolver os resultados.

4.4 *Gateway* de pagamentos

O sistema da *gateway* de pagamentos é um sistema transversal a qualquer organização podendo ser usado nos mais diversos sistemas. É responsável pela gestão e validação dos pagamentos que foram efectuados. Os pagamentos podem ser realizados de várias formas:

- ✓ Valores;
- ✓ Multibanco;
- ✓ Transferência bancária;
- ✓ Cartão crédito.

De modo a possibilitar fazer pagamentos de várias formas a *gateway* permite, não só carregar informação proveniente de instituições bancárias, como também registar o próprio acto do pagamento.

4.4.1 Arquitectura geral

A arquitectura da *gateway* foi construída com base na sua transversalidade. De seguida podemos ver a arquitectura geral da *gateway*, Figura 49.

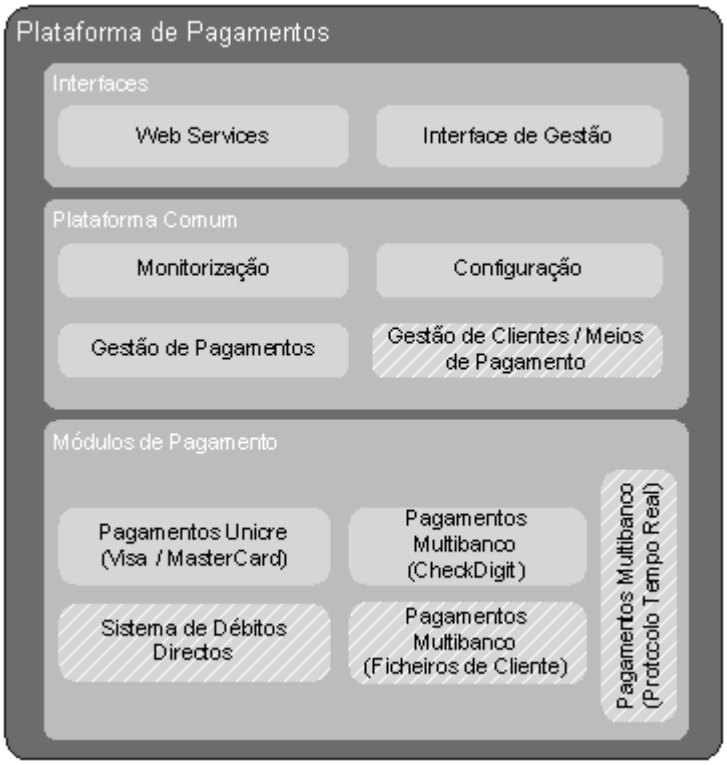


Figura 49 – Plataforma de pagamentos

Interfaces – Todas as aplicações interagem com a *gateway* através da invocação de *WebServices* próprios da *gateway*. A interface de gestão da plataforma, conforme já foi mencionado, serve para administração e configuração da *gateway*.

Plataforma Comum – A plataforma comum da *gateway* é utilizada para a configuração de contas de utilizadores e gestão dos pagamentos.

Módulos de pagamento – Os módulos de pagamentos são uma parte fundamental e complexa da *gateway* de pagamentos.

Cada um destes módulos implementa a integração com várias plataformas de pagamentos como a SIBS (Sociedade Interbancária de Serviços), no caso de pagamentos por cartão multibanco, ou com a UNICRE (pagamentos por cartão de crédito).

4.5 Plataforma Comum

Tendo em conta que o pedido de certidão do projecto RegD faz comunicação com a *gateway* de pagamentos, fiquei responsável pela plataforma comum da *gateway* para as outras aplicações que existiam.

O meu trabalho baseou-se em utilizar os *WebServices* disponibilizados para que as aplicações conseguissem interagir com a *gateway*, configuração de todos os módulos de pagamentos de modo a poderem ser utilizador.

De seguida é explicado como é efectuada a configuração dos módulos de pagamento mais complexos, por transferência bancária, por multibanco e por cartão de crédito.

4.5.1 Módulo de transferência bancária

O módulo de transferência bancária é responsável por disponibilizar uma interface que permite aos utilizadores do sistema registar todos os pagamentos que sejam realizados via transferência bancária (Figura 50). Este módulo também permite aos utilizadores actualizar a informação dos pagamentos registados na plataforma com base em informação contida em ficheiros provenientes de entidades bancárias (em formato csv; xml; tsv). Para isso acontecer a *gateway* permite fazer *upload* dos ficheiros e associar a informação contida nesses ficheiros.

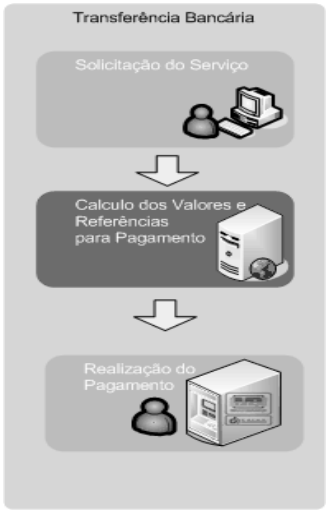


Figura 50 – Módulo de transferência bancária

Após a realização do pagamento por transferência bancária a entidade bancária emite um ficheiro (csv, xml ou tsv). Este ficheiro é disponibilizado pela entidade bancária e o cliente pode ir buscar o ficheiro dos pagamentos e fazer *upload* na *gateway* de pagamentos. Após o ficheiro correspondente ter sido carregado na *gateway* de pagamentos, todas as operações bancárias são analisadas para determinar se correspondem a pagamentos registados para transferência bancária. Estas operações bancárias são automaticamente associadas aos pagamentos correspondentes, Figura 51.

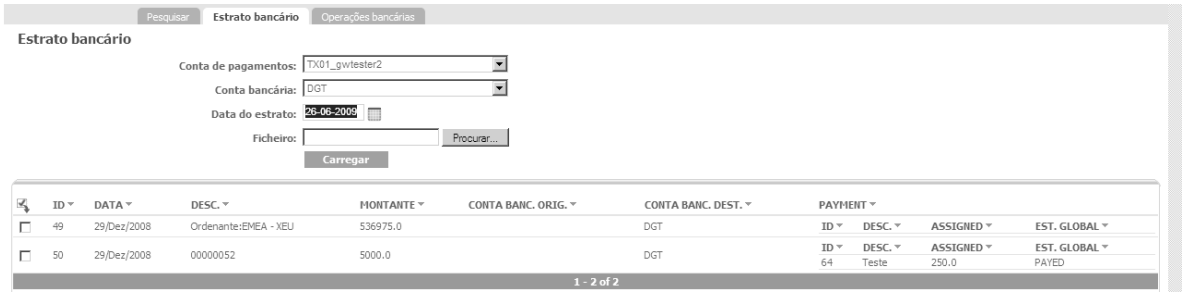


Figura 51 – Ecrã de carregamento de ficheiro

O ficheiro disponibilizado pela entidade bancária contém toda a informação relativa ao pagamento realizado. De seguida é apresentado um exemplo de uma linha de pagamento

```
06-05-2009 06-05-2009 Transf Conta-12345678(Banco:Nome) Ordenante 600 C 123456,87 123456,87
```

Esta linha contém os elementos do pagamento tais como:

- ✓ Data do movimento;
- ✓ Data da concretização do movimento;
- ✓ Tipo de transacção;
- ✓ Descritivo;
- ✓ Informação adicional (neste campo normalmente é definido a conta de origem);
- ✓ Valor da transacção;
- ✓ Transacção de crédito (C) ou débito (D);
- ✓ Saldo disponível;
- ✓ Saldo contabilístico.

Em anexo podemos ver um exemplo de ficheiro emitido pela entidade bancária. Para a *gateway* reconhecer os pagamentos que estão no ficheiro tem que ser configurada e essas configurações são realizadas através da interface de gestão da *gateway*, com acesso de administrador.

As configurações são constituídas por um par chave/valor. Seguidamente são apresentadas as várias chaves necessárias para a configuração da *gateway* no módulo de transferência bancária (Tabela 5).

Chave da configuração	Descrição
bank.reference.prefix	Prefixo adicionado a todas as referências para pagamento geradas.
bank.reference.suffix	Sufixo adicionado a todas as referências para pagamento geradas.
bank.reference.digits	Número de algarismos da referência gerada. O valor por omissão é 8.
bank.reference.putYear	Indicação se antes dos algarismos da referência deve ser colocado o ano.
bank.account.%.identifier	Identificador da conta bancária.
bank.account.%.description	Descrição da conta bancária.
bank.account.%.number	Número da conta bancária.
bank.account.%.file.firstOperationLine	Número de linha da primeira operação bancária no ficheiro do estrato bancário digital. O ficheiro só começará a ser processado à procura de operações bancárias a partir da linha indicada. Por omissão inicia na primeira linha.
bank.account.%.file.operationPattern	Padrão para extrair a informação de uma linha do ficheiro. Deve especificar o conteúdo de todas as colunas que deverão ser processadas, separadas por ";". Utiliza-se os *tokens* seguintes para especificar o conteúdo. "data:[FORMATO]" – data da operação no formato especificado "desc" – descrição da operação "valor" – valor da operação

	"tipo:C/D – tipo da operação, utilizando os símbolos para determinar sé é operação de crédito ou débito "orig" – identificação da origem da operação "dest" – identificação do destino da operação "ignora" – a coluna deve ser ignorada (pode ocorrer mais do que uma vez) Exemplo: "data:dd-MM-yyyy;ignora;desc;tipo:C,D;valor;ignora"
bank.account.%.file.type	Formato do ficheiro. São suportados os formatos: CSV, TSV e XLS.

Tabela 5 - Lista de configurações para transferência bancária

4.6 Módulo de multibanco

O módulo de multibanco implementa a integração com o Sistema de Pagamentos de Serviços / Compras da SIBS. Este é responsável por assegurar o processo de pagamento a partir do canal em questão. O processo de pagamento é realizado com recurso à utilização de *checkdigits*, permitindo assim às entidades fazerem pagamentos por cartão multibanco.

A SIBS cria diariamente um ficheiro de movimentos (do tipo .inp) que contém todos os pagamentos efectuados durante o dia anterior. Este ficheiro é disponibilizado através de uma ligação *dial-up* dedicada e a tesouraria do cliente pode fazer *download* do ficheiro e carregá-lo na *gateway*. Este ficheiro pode também ser criado três vezes por dia, dependendo da actividade do cliente.

O ficheiro é constituído pelos seguintes tipos de registo e apresentam-se ordenados do seguinte modo:

- ✓ Tipo de registo 0 – *header* do ficheiro;
- ✓ Tipo de registo 2 – detalhe de cada registo no ficheiro;
- ✓ Tipo de registo 9 – *trailer* do ficheiro.

Cada linha do ficheiro é constituída exactamente por 100 caracteres. Em anexo podemos ver um exemplo completo do ficheiro de movimentos.

De seguida é apresentado um exemplo de uma linha de um movimento bancário.

```
2040012000004562009022009350000010000025 00AT000034579202740        LISBOA000016482
```

Cada número ou conjunto de números tem a sua correspondência (Tabela 6).

Descrição	Caracteres	Valor de exemplo
Tipo registo	1	=2
Código processamento	2	=04
Identificação log SIBS	4	=0012
Num.Log SIBS	8	=00000456
Data/hora transacção cliente	12	=200902200935
Montante pago	10	=0000010000
Tarifa	5	=02500
Tipo de terminal	2	=AT
Identificação do terminal	10	=0000345792
Identificação da transacção	5	=02740
Localidade do terminal	15	= LISBOA
Referência do pagamento	9	=000016482
Modo de envio	1	= não observado
Cod. Resposta da empresa	1	= não observado
Número Identificação resposta	12	= não observado

Tabela 6 – Descrição do conteúdo do ficheiro

Para a *gateway* reconhecer este formato de ficheiro é necessário fazer a respectiva configuração (Tabela 7).

Chave da configuração	Descrição
mb.sibs.entidade	Código da entidade utilizado nas operações por multibanco.
mb.sibs.instituicao	Identificativo da instituição no serviço multibanco.

Tabela 7 – Configuração do módulo de multibanco

Para fazer o *upload* do ficheiro é necessário aceder à interface de gestão da *gateway* e carregar o ficheiro na aplicação (Figura 52).

Figura 52 – Ecrã de carregamento de ficheiro

Todos os pagamentos correspondentes a entradas no ficheiro são considerados pagos se tiverem a correspondência com pagamentos criados na *gateway* de pagamentos. Essa correspondência é feita automaticamente não podendo haver alterações manuais (como pode acontecer com o módulo de transferências bancárias).

4.7 Cartão de crédito

O módulo de cartão de crédito é mais complexo da *gateway* pois ocorre em *real-time* (pagamento realizado no momento em que é submetido) e contém vários pontos de segurança para se poder realizar. Este módulo implementa a integração com a UNICRE.

Na Figura 53 podemos observar o processo do pagamento por cartão de crédito. O último passo, na conclusão de pagamento, é o mais importante e complexo de todo o ciclo do processo de pagamento por cartão de crédito.

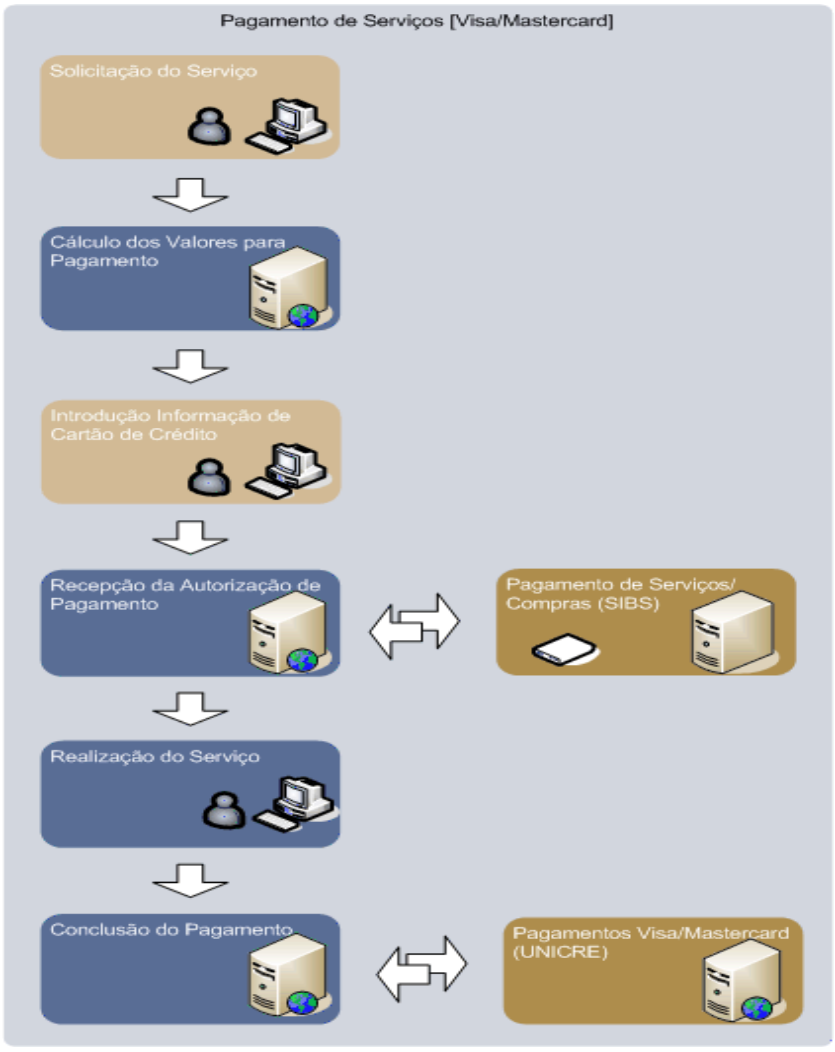

Pagamento de Serviços [Visa/Mastercard]
Solicitação do Serviço
Cálculo dos Valores para Pagamento
Introdução Informação de Cartão de Crédito
Recepção da Autorização de Pagamento
Pagamento de Serviços/ Compras (SIBS)
Realização do Serviço
Conclusão do Pagamento
Pagamentos Visa/Mastercard (UNICRE)

Figura 53 – Processo de pagamento por cartão de crédito

A imagem seguinte (Figura 54) esclarece a troca de informação que existe na fase de conclusão do pagamento.

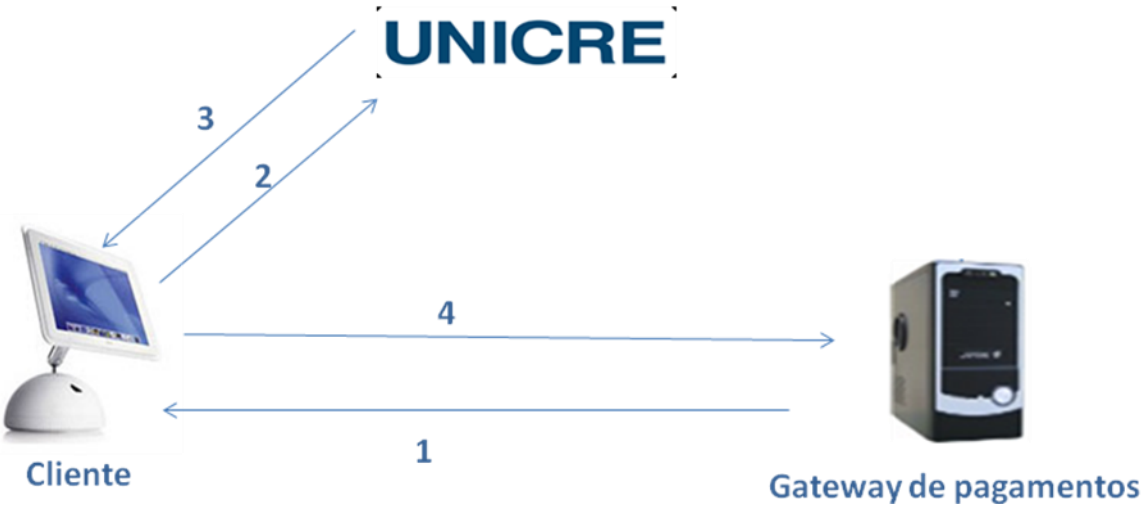


Figura 54 – Troca de mensagens

Na fase de conclusão do pagamento temos quatro passos principais:

- ✓ 1 – A *gateway* devolve ao cliente/sistema os dados do pagamento;
- ✓ 2 – O cliente/sistema cria um HTML baseado no protocolo *Form 3*. Na sequência de processamento esta será a primeira mensagem a ser enviada, permitindo a autenticação do cartão de crédito. Neste pedido também vai um URL de retorno de modo a que quando a resposta vier da UNICRE o browser do cliente/sistema saiba para onde vai ser redireccionado;
- ✓ 3 – A UNICRE devolve a resposta em HTML de que o cartão foi aceite ou não e juntamente com a resposta envia um URL com o retorno para a página onde o cliente/sistema estava;
- ✓ 4 – Por último a informação de que o pagamento foi ou não efectivado será enviada para a *gateway* de modo a dar como concluído o ciclo do processo do módulo de pagamento por cartão de crédito.

Todos os pedidos entre o cliente/sistema, *gateway* e UNICRE são realizados através um canal seguro de HTTPS (SSL). Para isso é necessário recorrer a dois certificados digitais. Um certificado digital da UNICRE, para a comunicação do cliente/sistema para a UNICRE (este certificado é garantido pela UNICRE) e um certificado digital para a comunicação entre o cliente/sistema e a *gateway* de pagamentos. Este certificado digital será emitido por uma autoridade de certificação. Para a fase de testes criou-se um certificado digital de testes de modo a conseguir simular todo o processo, instalou-se o certificado de testes na *gateway* e teve que se configurar os servidores *MyWebMethods* para aceitarem este certificado.

Além da configuração do certificado existem outras configurações para serem feitas de modo a poder concretizar o pagamento por cartão de crédito (Tabela 7).

Chave da configuração	Descrição
card.unicre.tpav.number	Número do TPA (Terminais de Pagamento Automático) virtual.
card.unicre.url.authentication	URL para efectuar a autenticação na UNICRE.
card.unicre.url.message	URL para envio de mensagens para a UNICRE.
card.unicre.url.message	*Hash* criptográfico utilizado na autenticação.
card.unicre.secureHash	Certificado para autenticação perante a UNICRE.

card.unicre.certificate	Palavra passe de acesso ao certificado.
card.unicre.certificate.password	Número do TPA virtual.

Tabela 8 – Configuração do módulo de cartão de crédito

Nestas configurações são definidos o URL para a autenticação e envio de mensagens para a UNICRE e o certificado para o acesso a UNICRE.

Capítulo 5 Conclusão

Neste capítulo será apresentado um resumo das dificuldades encontradas e das competências adquiridas. Por fim apresenta-se o trabalho futuro, a nível do projecto RegD assim como dos novos desafios na *Indra*.

5.1 Dificuldades encontradas e competências adquiridas

No início do estágio realizado na *Indra* foram definidos os objectivos, os quais se encontram apresentados no Capítulo 1. Todos estes objectivos foram cumpridos e foram inclusivamente acrescentados novos objectivos.

Durante todo o estágio adquiri um grande conjunto de competências tanto a nível profissional como a nível pessoal. Com a ajuda da *Indra* (com o projecto "Integrate"), bem como da excelente equipa do projecto, consegui que a adaptação ao ambiente profissional fosse excelente. Foi definitivamente um factor importante para o sucesso e concretização do meu estágio.

Ao longo da realização do estágio encontrei alguns obstáculos. Esses obstáculos acorreram por diversos motivos, alguns associados ao complexo processo de negócio que envolvia o projecto, outros provenientes das ferramentas utilizadas. Todos estes obstáculos contribuíram também para a minha aprendizagem, bem como para aprofundar os conhecimentos adquiridos na faculdade.

5.2 Trabalho futuro

O projecto RegD e a *gateway* de pagamentos, neste momento, encontram-se em fase inicial de produção. Faltam algumas configurações nos servidores de produção do RegD e posteriormente serão realizados testes de despiste de modo a garantir que a passagem total para produção ocorre com sucesso.

Quando o projecto estiver totalmente em produção, alguns elementos da equipa de projecto irão permanecer no cliente para dar apoio ao projecto e corrigir alguma

incidência que ocorra. Actualmente alguns elementos da equipa estão a ser alocados em novos projectos.

Após o estágio curricular, permaneci no projecto RegD, e continuei a dar suporte na passagem da *gateway* de pagamentos para produção. Após um mês de conclusão do estágio fui alocado noutro projecto continuando a ganhar novos conhecimentos e competências.

Capítulo 6 Bibliografia

[1] SoftwareAG, WebMethods Developer User's Guide Version, Janeiro de 2008.

[2] SoftwareAG, WebMethods Installation guide, Janeiro de 2008

[3] Indra, Documentação interna Indra Sistemas de Portugal, 2010

[4] Rui Almeida, João Veiga, Rute Arez, Manual de instação do Registo de Dispositivos, 2011.

[5] Susana Estevens, João Veiga, Rute Arez, Especificação Funcional Milestone 2 - Pedido de Certidão, 2010.

[6] Rui Almeida, João Veiga, Rute Arez, Especificação Funcional Release 1 - Registo de Fabricantes, 2011.

[7] Rui Almeida, Manual de Desenvolvimento – Gateway de pagamentos, 2011.

[8] Alexandre Pinto, Nuno Dias, Manual de instalação – Gateway de pagamentos 2010.

[9] SIBS, Produto e-commerce 3D secure/VISA Processamento de transacções não presenciais, 2007.

[10] SIBS, Produto e-commerce 3D secure/VISA Manual de Apoio a comerciantes, 2007.

[11] WebMethods

www.softwareag.com/corporate/default.asp

en.wikipedia.org/wiki/WebMethod

searchsoa.techtarget.com/definition

[12] WebMethods IntegrationServer

en.wikipedia.org/wiki/WebMethods_Integration_Server

[13] WebMethods Designer

www.wmusers.com/forum/showthread.php?t=13888

communities.softwareag.com/ecosystem/communities/public/Developer/webmethods/tutorials/CAF/CAFandJSF/JSF_Used-in-CAF-App.html

documentation.softwareag.com/webmethods/wmsuites/wmsuite7/relnotes/webMethods%20Product%20Suite%20Release%20Notes%207.2.htm

[14] Redunicre

www.redunicre.pt

[15] Unicre

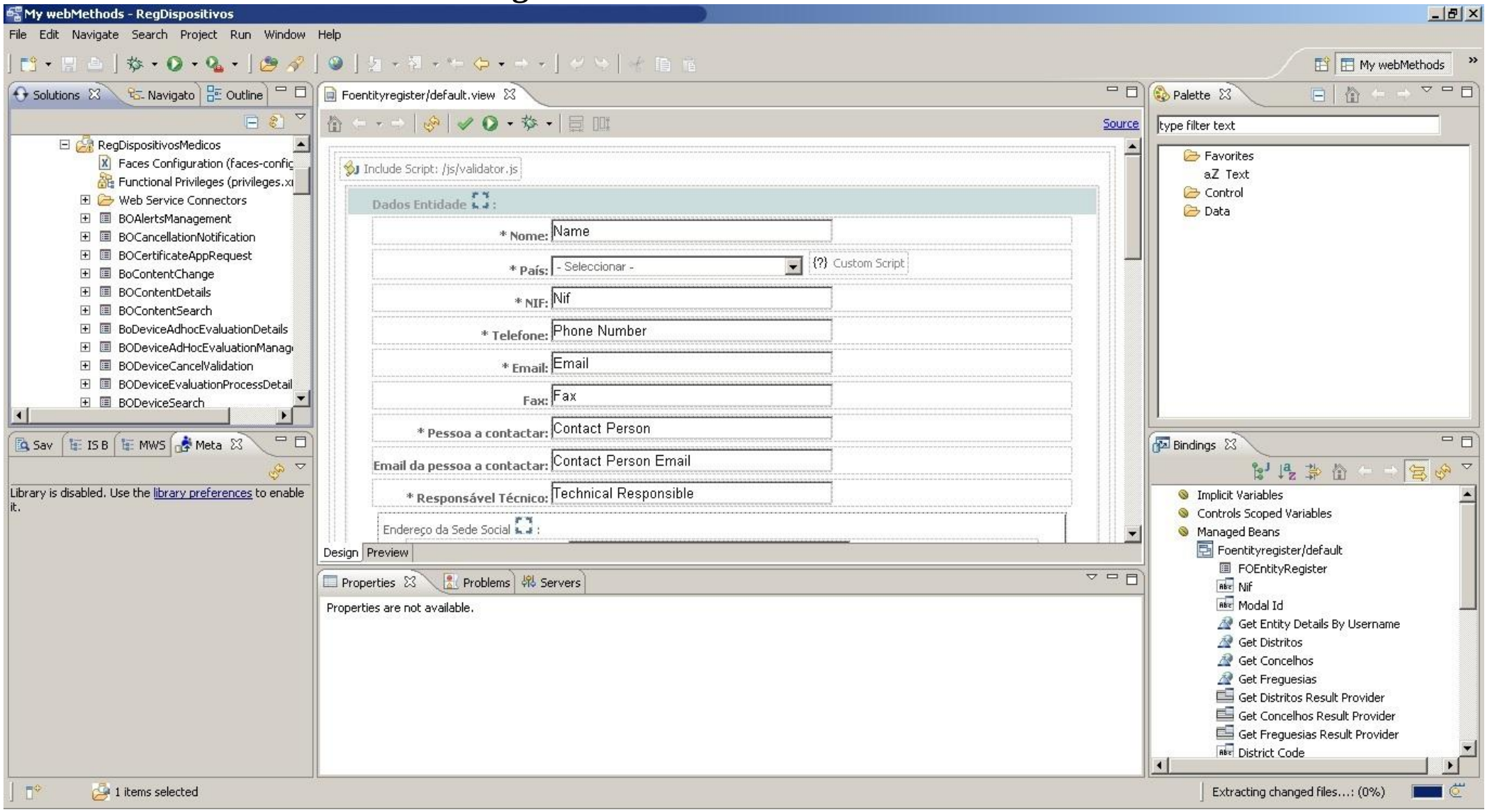
www.unicre.pt

[16] SIBS

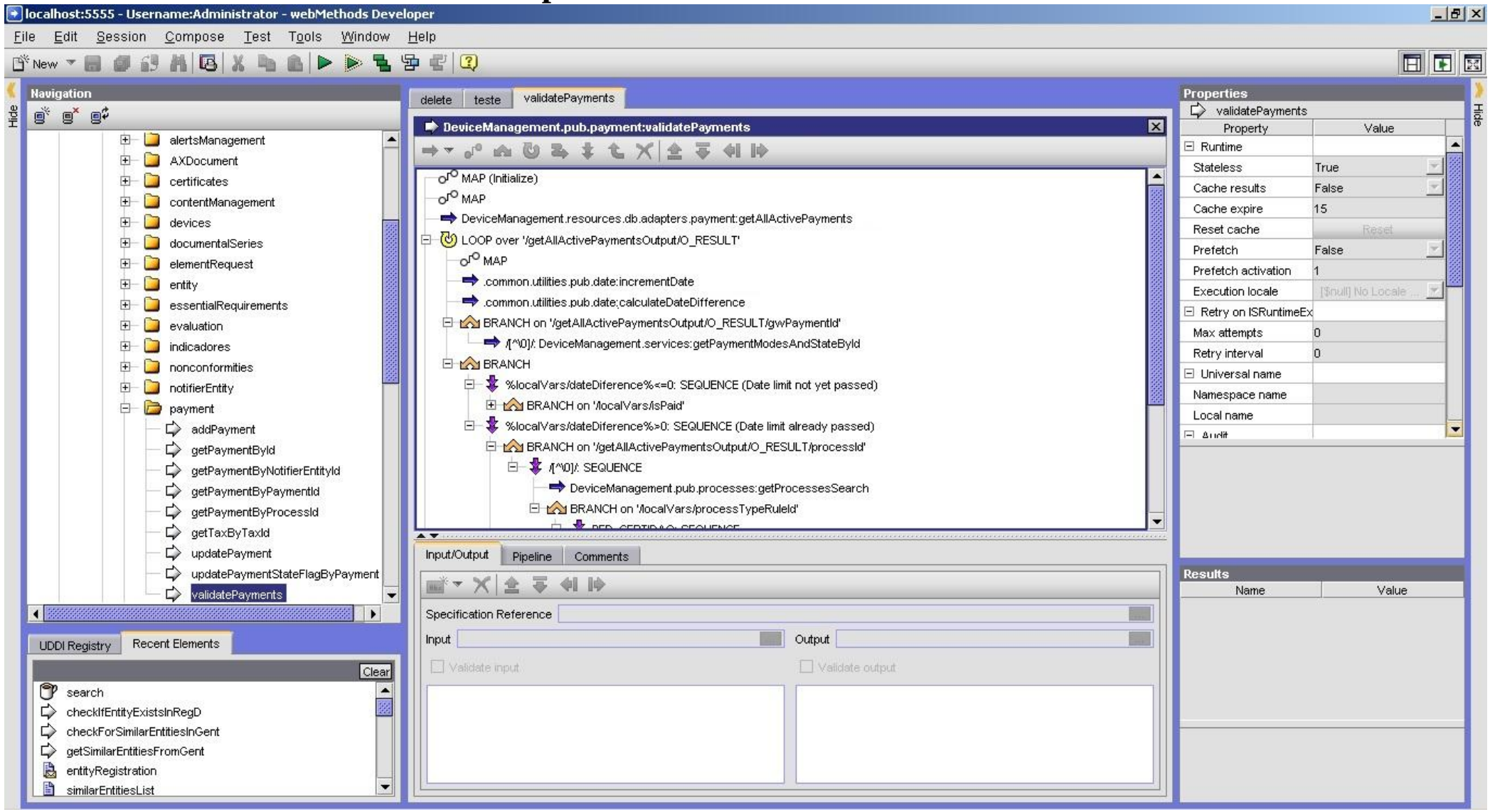
www.sibs.pt/pt

websites acedidos em Maio de 2011

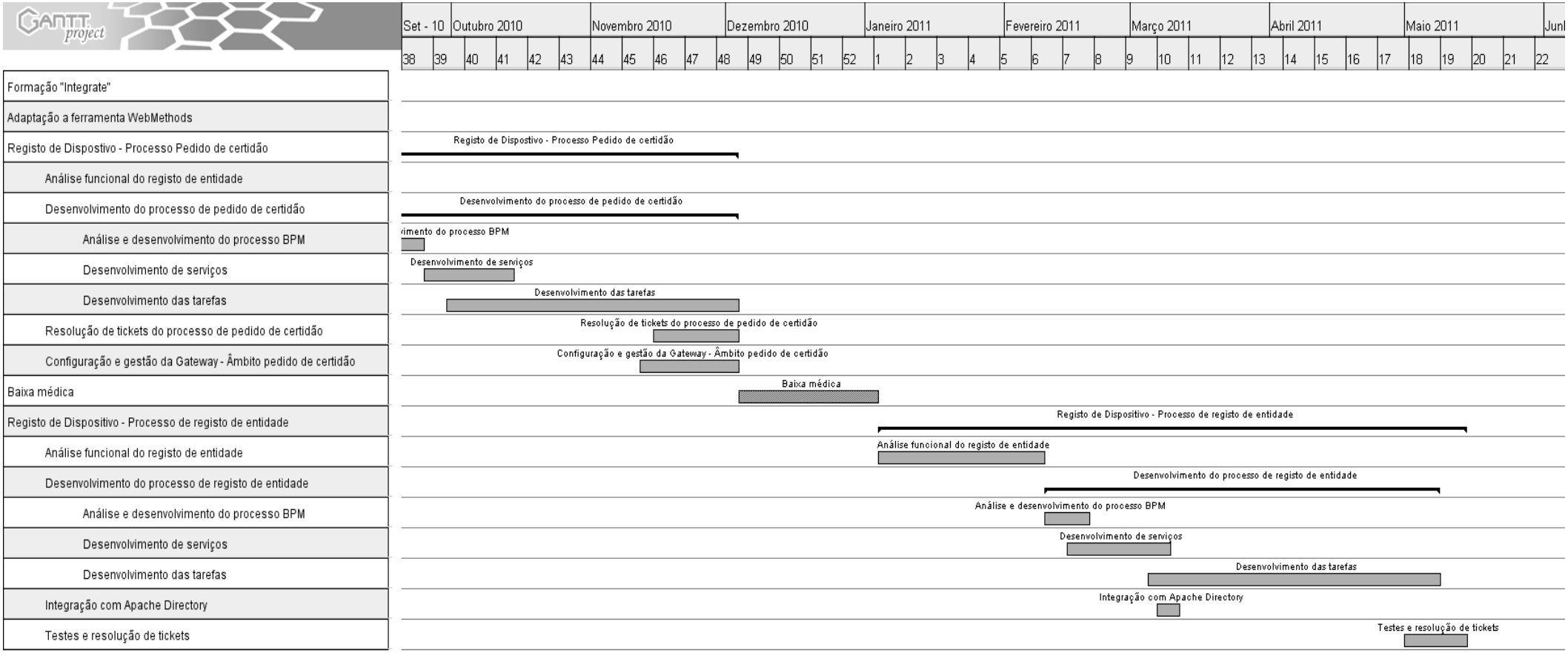
Anexo – WebMethods Designer



Anexo – WebMethods Developer



Anexo – WebMethods Developer


GANTT project
Set - 10
Outubro 2010
Novembro 2010
Dezembro 2010
Janeiro 2011
Fevereiro 2011
Março 2011
Abril 2011
Maio 2011
38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
Formação "Integrate"
Adaptação a ferramenta WebMethods
Registo de Dispostivo - Processo Pedido de certidão
Análise funcional do registo de entidade
Desenvolvimento do processo de pedido de certidão
Análise e desenvolvimento do processo BPM
Desenvolvimento de serviços
Desenvolvimento das tarefas
Resolução de tickets do processo de pedido de certidão
Configuração e gestão da Gateway - Âmbito pedido de certidão
Baixa médica
Registo de Dispositivo - Processo de registo de entidade
Análise funcional do registo de entidade
Desenvolvimento do processo de registo de entidade
Análise e desenvolvimento do processo BPM
Desenvolvimento de serviços
Desenvolvimento das tarefas
Integração com Apache Directory
Testes e resolução de tickets

Anexo - Ficheiro de Transferência Bancária

Organismo: AUTORIDADE NACIONAL DOS DISPOSITIVOS

Responsável: Direcção

N° Conta: 2312 AUTORIDADE Taxa 14

NIB: 2222 3333 12312312312 89

Resp. Direcção

Telefone: 222 222 222

Email: rebrito@indracompany.com

Interlocutor:Rui Almeida

Moeda Visualização: EUR

	Disponível	Contabilistico
Saldo Inicial:	**393395,92**	**393275,56**

Data Mov.	Data Valor	Tipo	Descritivo	Informação Adicional	Valor em EUR	Saldo Disponivel	Saldo Contabilistico

02-02-2009	2009/02/02	TransferênciaTransferência da Conta N° 2222(Banco: BES)	Ordenante:A J COSTA LDA	239,41	C	391435,33	392914,97				
02-02-2009	2009/02/02	TransferênciaTransferência da Conta N° 123123123(Banco: BST)	Ordenante:PAUL HARTMANN LDA	189,97	C	391625,3	393104,94				
12-02-2009	2009/02/11	Outros	Depósito de Numerário Externo	Depósito de numerário através dos Depósitos Externos	5,04	C	429531,56	429531,56			
12-02-2009	2009/02/11	Cheque	Depósito do Cheque Externo N° 2234324234	Cheque depositado através dos Depósitos Externos	2148,3	C	429531,56	431679,86			
17-02-2009	2009/02/17	TransferênciaTransferência da Conta N° 231274094(Banco: BCP)	PORTU	7,74	C	438343,11	438343,11				
17-02-2009	2009/02/17	TransferênciaTransferência da Conta N° 234234(Banco: CGD)	Ordenante:OLYMPUS	783,39	C	439126,5	439126,5				
18-02-2009	2009/02/18	TransferênciaTransferência da Conta N° 23423432(Banco: BES)	Ordenante:SANO TECNICA LDA	216,28	C	439342,78	439342,78				
18-02-2009	2009/02/18	TransferênciaTransferência da Conta N° 234234324(Banco: CGD)	Ordenante: CONSULTADOR	1116,32	C	440459,1	440459,1				
20-02-2009	2009/02/20	TransferênciaTransferência da Conta N° 234324234(Banco: BCP)	Ordenante: DIST PROD HOS	14	C	440473,1	440473,1				
12-02-2009	2009/02/11	Cheque	Depósito do Cheque Externo N° 555235232	Cheque depositado através dos Depósitos Externos	460,35	C	429531,56	432140,21			
12-02-2009	2009/02/11	Cheque	Depósito do Cheque Externo N° 2026328809	Cheque depositado através dos Depósitos Externos	102,3	C	429531,56	432242,51			
12-02-2009	2009/02/11	Cheque	Depósito do Cheque Externo N° 3709022901	Cheque depositado através dos Depósitos Externos							

```
      102,3  C      429531,56     432344,81

12-02-2009    2009/02/11    Cheque Depósito do Cheque Externo N° 7503432042      Cheque depositado através dos Depósitos Externos
      102,3  C      429531,56     432447,11

12-02-2009    2009/02/11    Cheque Depósito do Cheque Externo N° 7603432085      Cheque depositado através dos Depósitos Externos
      102,3  C      429531,56     432549,41

12-02-2009    2009/02/11    Cheque Depósito do Cheque Externo N° 4371721226      Cheque depositado através dos Depósitos Externos
      102,3  C      429531,56     432651,71

13-02-2009    2009/02/13    TransferênciaTransferência da Conta N° 234234234(Banco: DBANK)    Ordenante:GASES    INDUSTRIAIS    S.A.
      18,08  C      429872,87     437842,98

13-02-2009    2009/02/11    Cheque Compensação do Cheque Externo N° 234234324    Cheque depositado através dos Depósitos Externos
      2148,3 C      432021,17     437842,98

26-02-2009    2009/02/26    TransferênciaTransferência da Conta N° 123412341234(Banco: BBPI)  Ordenante:VITALINO L       177,23 C
      450113,3      450113,3

26-02-2009    2009/02/26    TransferênciaTransferência da Conta N° 123412341(Banco: BARCL)    Ordenante:BAUSCH S.A.      467,8  C
      450581,1      450581,1

26-02-2009    2009/02/26    TransferênciaTransferência da Conta N° 12341234(Banco: BARCL)     Ordenante:SMITHS            (PORTUGAL)
      91,55  C      450672,65     450672,65

26-02-2009    2009/02/26    TransferênciaTransferência da Conta N° 12341234(Banco: BCP)       Ordenante:ST PR     10     C
      450682,65     450682,65

26-02-2009    2009/02/26    TransferênciaTransferência da Conta N° 12341234(Banco: BCP)       Ordenante:ST DIS PR 10     C
      450692,65     450692,65 13-02-2009

2009/02/11    Cheque Compensação do Cheque Externo N° 234324234    Cheque depositado através dos Depósitos Externos     102,3  C
      432890,72     437842,98

13-02-2009    2009/02/11    Cheque Compensação do Cheque Externo N° 234324234    Cheque depositado através dos Depósitos Externos
```

102,3 C 432993,02 437842,98

13-02-2009 2009/02/11 Cheque Compensação do Cheque Externo N° 23432432423 Cheque depositado através dos Depósitos Externos
102,3 C 433095,32 437842,98

13-02-2009 2009/02/11 Cheque Compensação do Cheque Externo N° 234234324234 Cheque depositado através dos Depósitos Externos
808,4 C 433903,72 437842,98

Disponível Contabilistico

Saldo Final: 451423,32 451423,32

Anexo - Ficheiro de Movimentos por Multibanco

```
0MEPS500000009008296720090302120090320311242978 20
20400120000045620090220093500000100000250 0AT000034579202740         LISBOA000015136
20400120000045620090220094000000075000250 0AT000034579202740         LISBOA000015218
20400120000045620090220095000000100000250 0AT000034579202740         LISBOA000015136
20400120000045620090220095500000576560250 0AT000034579202740         LISBOA000015348
900000004000000000000851560000000100000000000000145
```